



US006493749B2

(12) **United States Patent**  
**Paxhia et al.**

(10) Patent No.: **US 6,493,749 B2**  
(45) Date of Patent: **\*Dec. 10, 2002**

(54) **SYSTEM AND METHOD FOR AN ADMINISTRATION SERVER**(75) Inventors: **Frank V. Paxhia, Binghamton, NY (US); Kurt A. Strelfert, Vestal, NY (US)**(73) Assignee: **International Business Machines Corporation, Armonk, NY (US)**

(\*) Notice: This patent issued on a continued prosecution application filed under 37 CFR 1.53(d), and is subject to the twenty year patent term provisions of 35 U.S.C. 154(a)(2).

Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/135,147**(22) Filed: **Aug. 17, 1998**(65) **Prior Publication Data**

US 2002/0052935 A1 May 2, 2002

(51) **Int. Cl. 7** ..... **G06F 15/173**(52) **U.S. Cl.** ..... **709/220; 345/735; 709/223**(58) **Field of Search** ..... **709/220, 221, 709/222, 223, 226; 345/735-6**(56) **References Cited**

## U.S. PATENT DOCUMENTS

- |               |         |                   |       |         |
|---------------|---------|-------------------|-------|---------|
| 5,022,077 A * | 6/1991  | Bealkowski et al. | ..... | 380/4   |
| 5,475,819 A * | 12/1995 | Miller et al.     | ..... | 709/203 |
| 5,495,607 A   | 2/1996  | Pisello et al.    | ..... | 395/600 |
| 5,555,375 A * | 9/1996  | Sudama et al.     | ..... | 709/226 |
| 5,734,831 A * | 3/1998  | Sanders           | ..... | 709/223 |
| 5,740,431 A * | 4/1998  | Rail              | ..... | 707/100 |
| 5,771,381 A * | 6/1998  | Jones et al.      | ..... | 713/100 |
| 5,867,706 A * | 2/1999  | Martin et al.     | ..... | 709/105 |
| 5,872,928 A * | 2/1999  | Lewis et al.      | ..... | 709/222 |

5,875,327 A *	2/1999	Brandt et al.	.....	713/1
5,991,808 A *	11/1999	Broder et al.	.....	709/226
6,076,106 A *	6/2000	Hamner et al.	.....	709/223
6,078,955 A *	6/2000	Konno et al.	.....	709/224
6,102,965 A *	8/2000	Dye et al.	.....	717/1
6,105,063 A *	8/2000	Hayes, Jr.	.....	709/223
6,182,075 B1 *	1/2001	Hsu	.....	707/10
6,279,001 B1 *	8/2001	DeBettencourt et al.	.....	707/10

## OTHER PUBLICATIONS

Sant'Angelo et al., *Windows NT Server Survival Guide*, SAMS Publishing, 1996, p. 303.\*  
Koster, *Archie-Like Indexing in the WEB*, Comput. Netw. ISDN Syst., vol. 27, No. 2, pp. 175-182, Nov. 1994. INSM AN 4842959.

Voegele, et al. *Slinky: a URL server*, Proceedings. The First Annual Conference on Emerging Technologies and Applications in Communications, IEEE Comput. Soc. Press, pp. 120-123, 1996. INSM AN 5335261.

\* cited by examiner

Primary Examiner—Dung C. Dinh

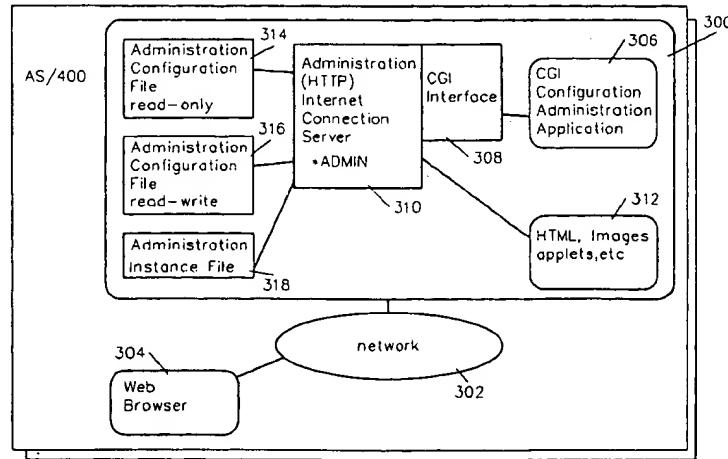
Assistant Examiner—Bradley Edelman

(74) Attorney, Agent, or Firm—Shelley M Beckstrand

(57) **ABSTRACT**

System and method for serving HTML pages to web browsers for the purpose of administration and configuration. A plurality of instances of WWW servers is provided, with one such instance including a configuration file which is restricted in usage and not alterable by way of any HTML configuration or administration forms. This plurality of instances of internet connection servers is managed by way of a web browser which displays and interacts with a plurality of HTML forms and corresponding common gateway interface binary programs which are provided selectively for creating and deleting instances of servers, associating a configuration file with a server instance, changing server instance start up parameters, and starting, ending, and restarting server instances.

**15 Claims, 19 Drawing Sheets**



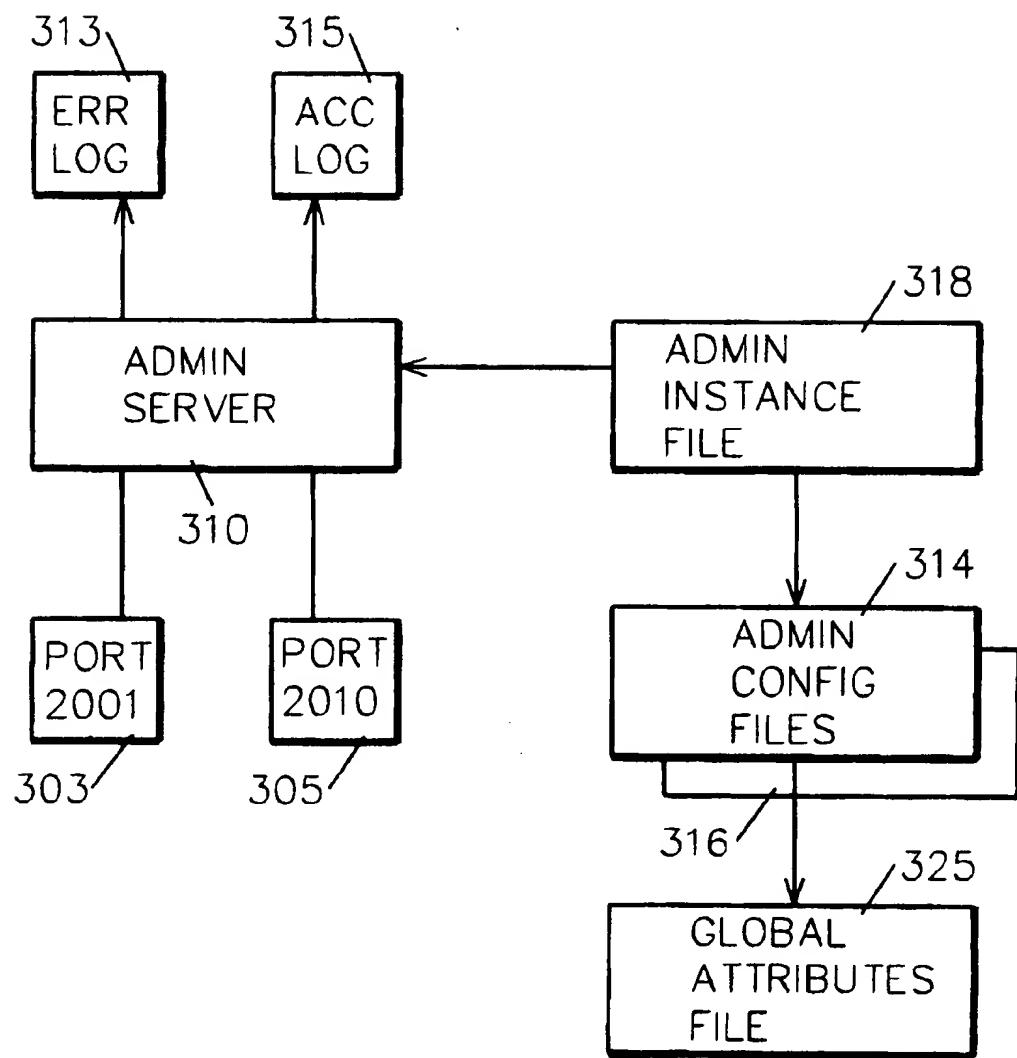


FIG. 1

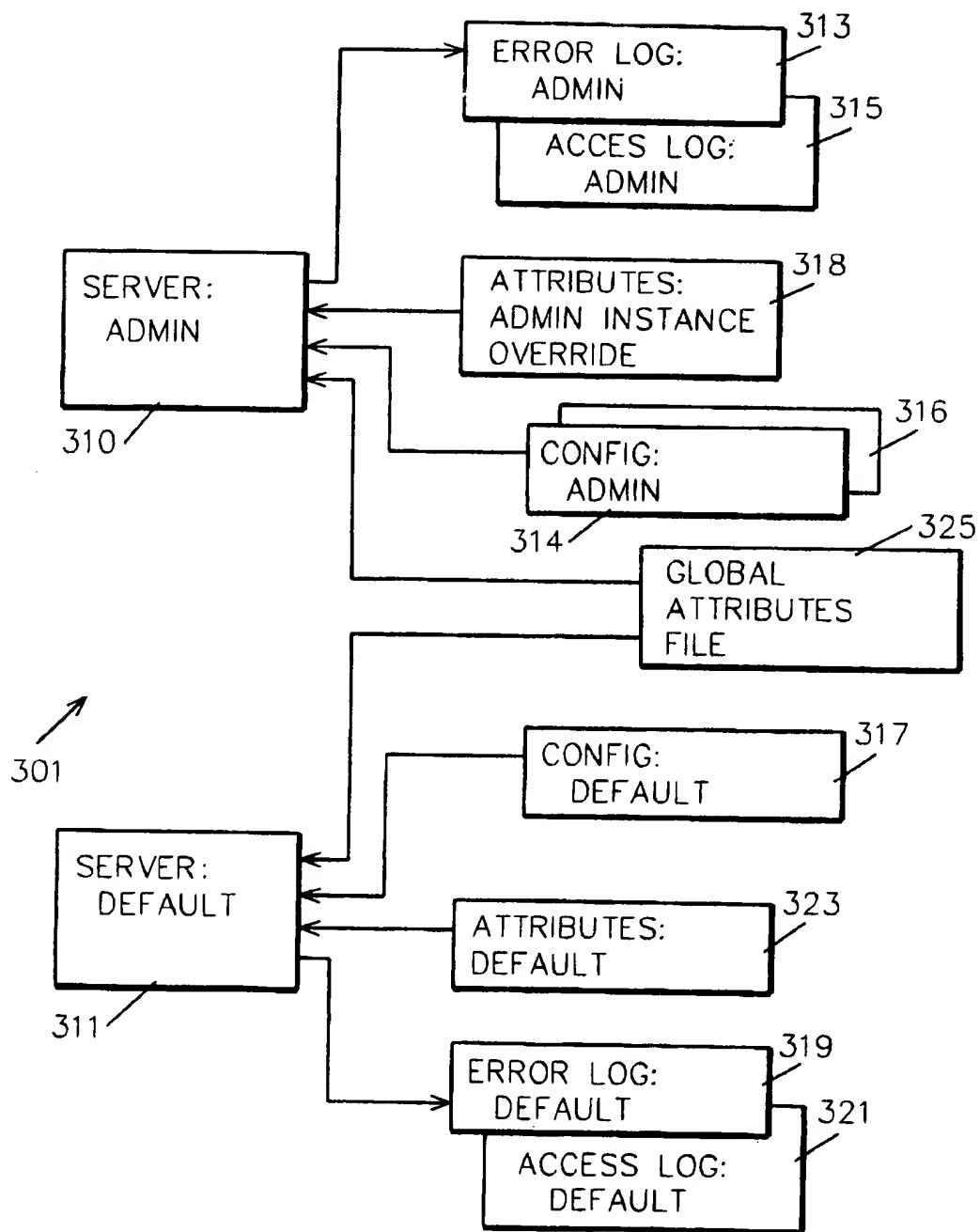


FIG. 2

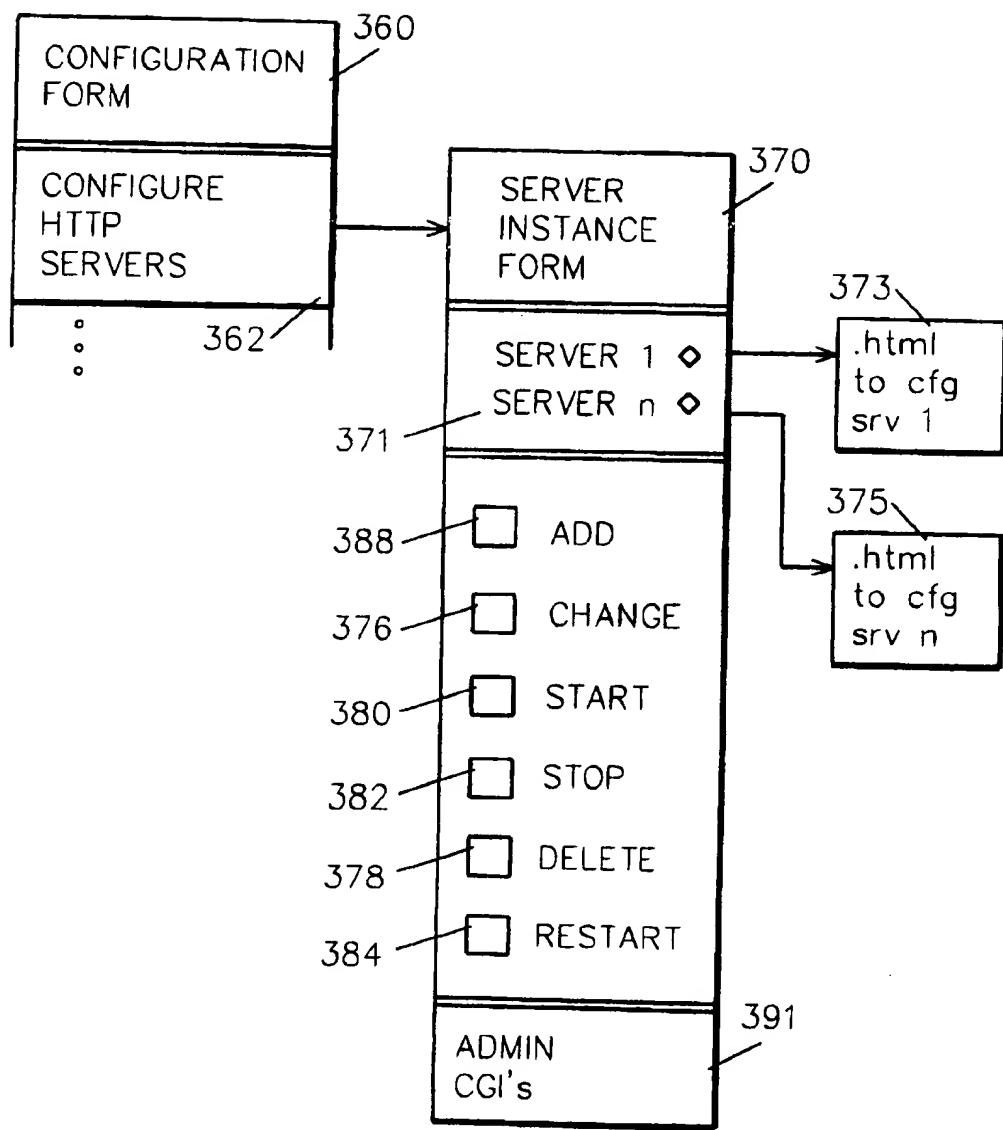


FIG. 3

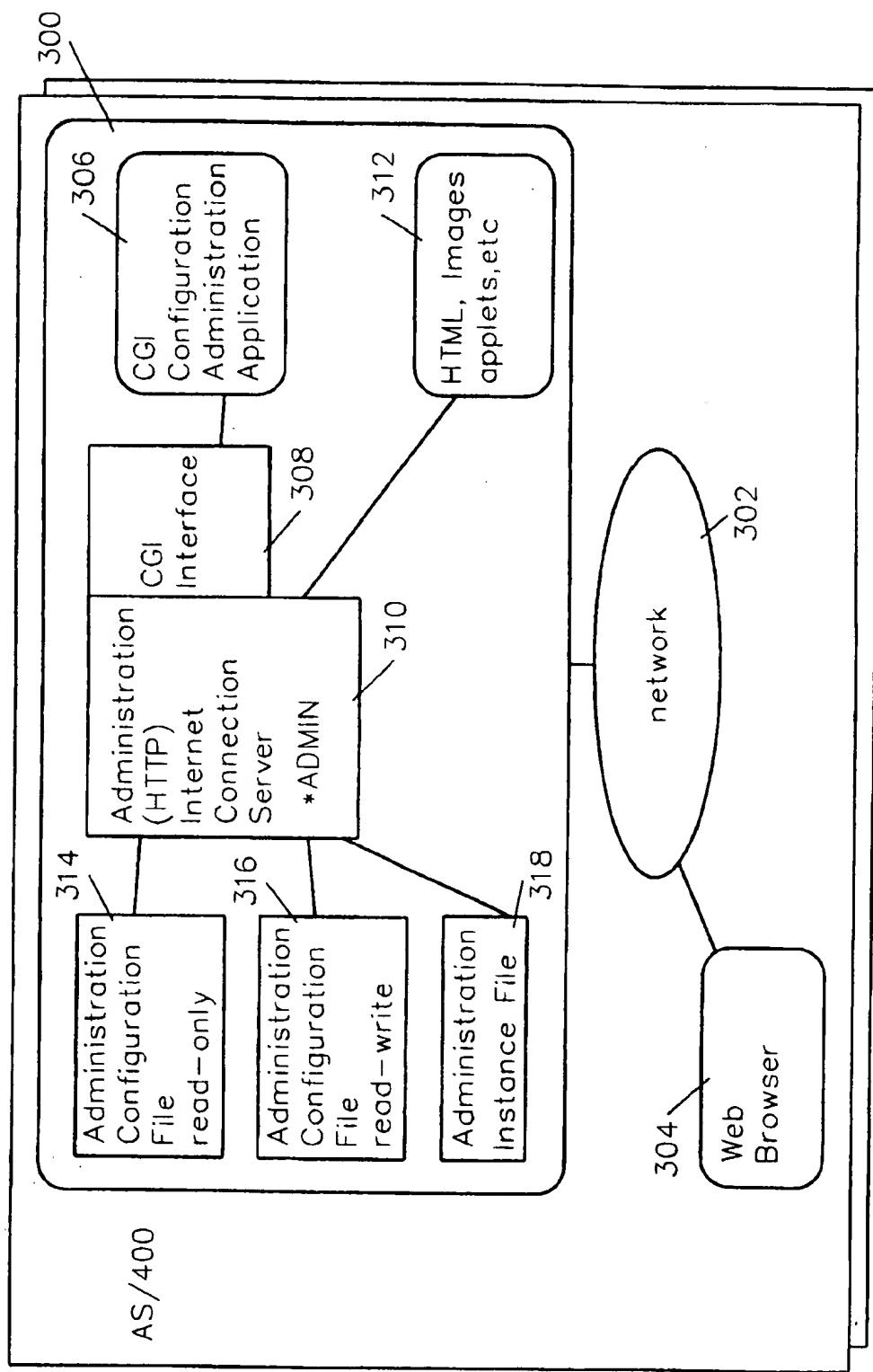
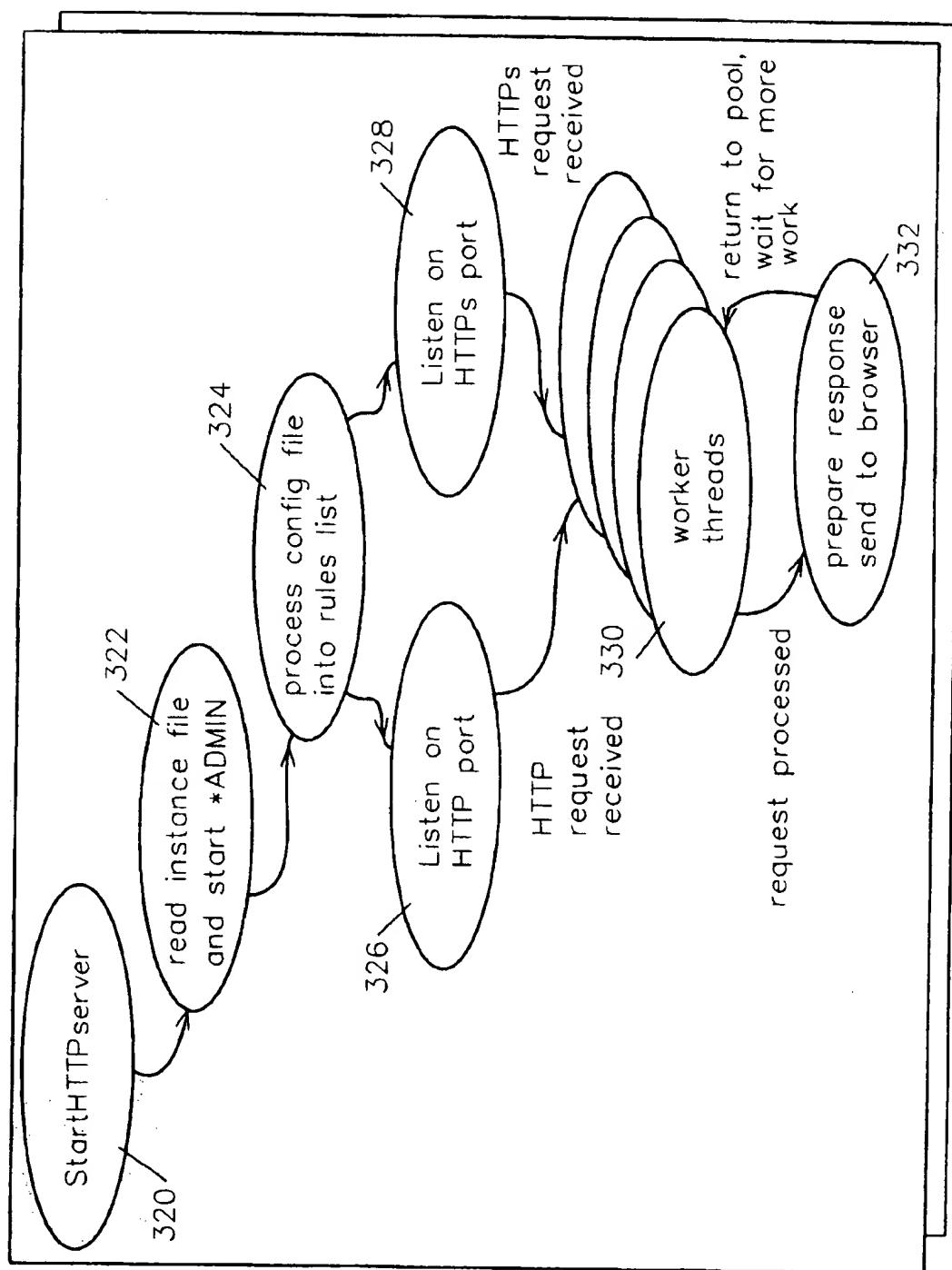


FIG. 4

FIG. 5



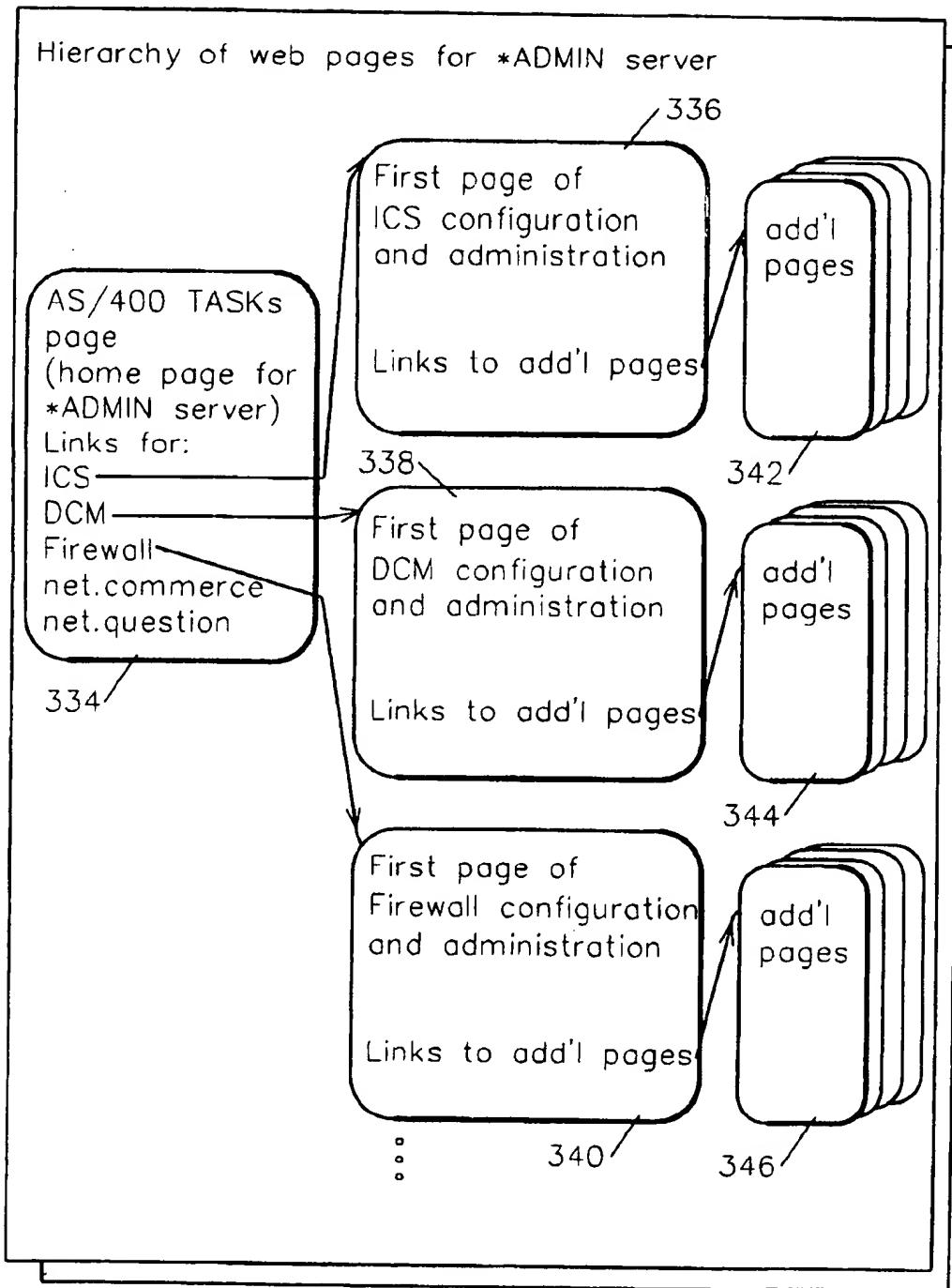


FIG. 6

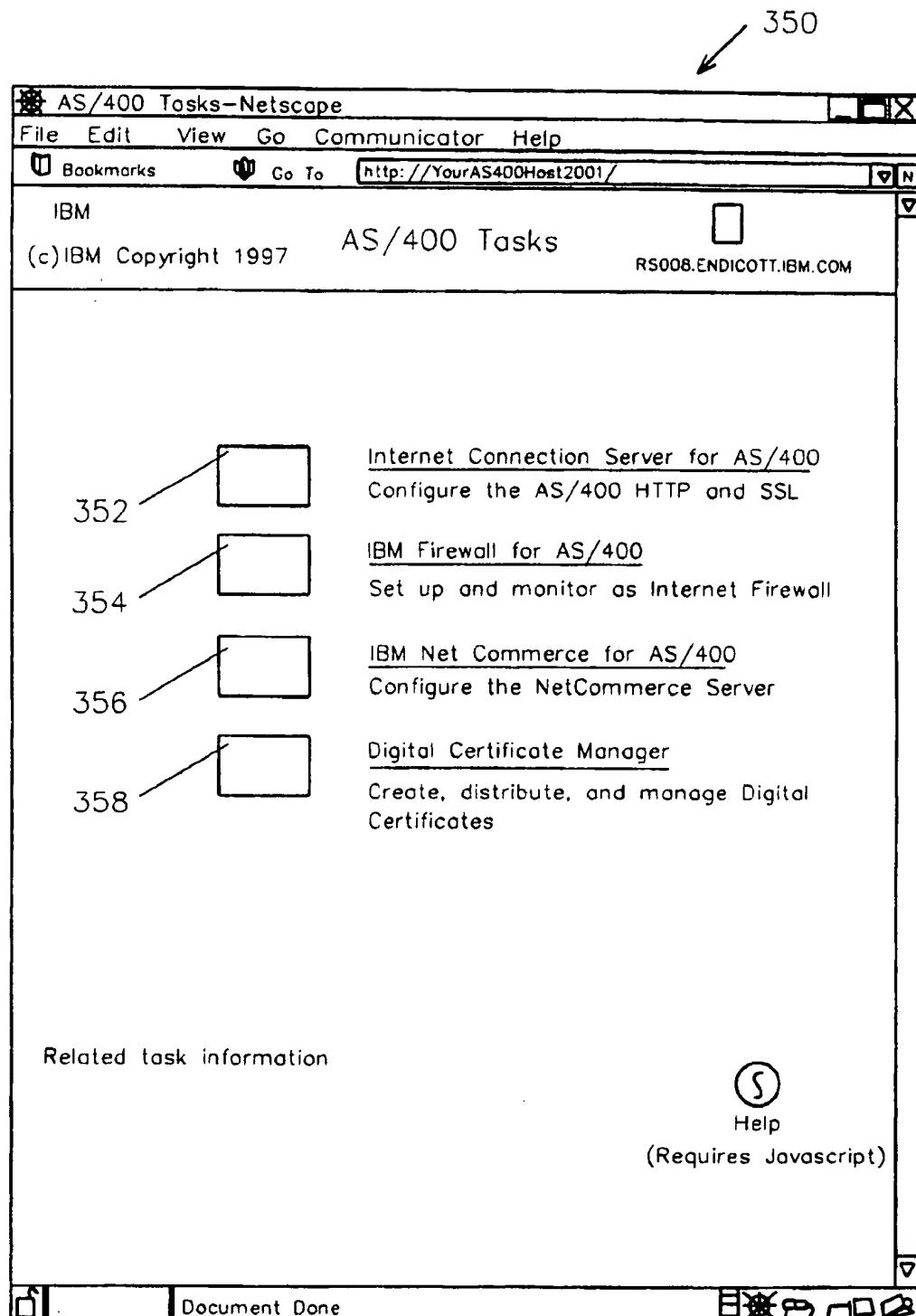


FIG. 7

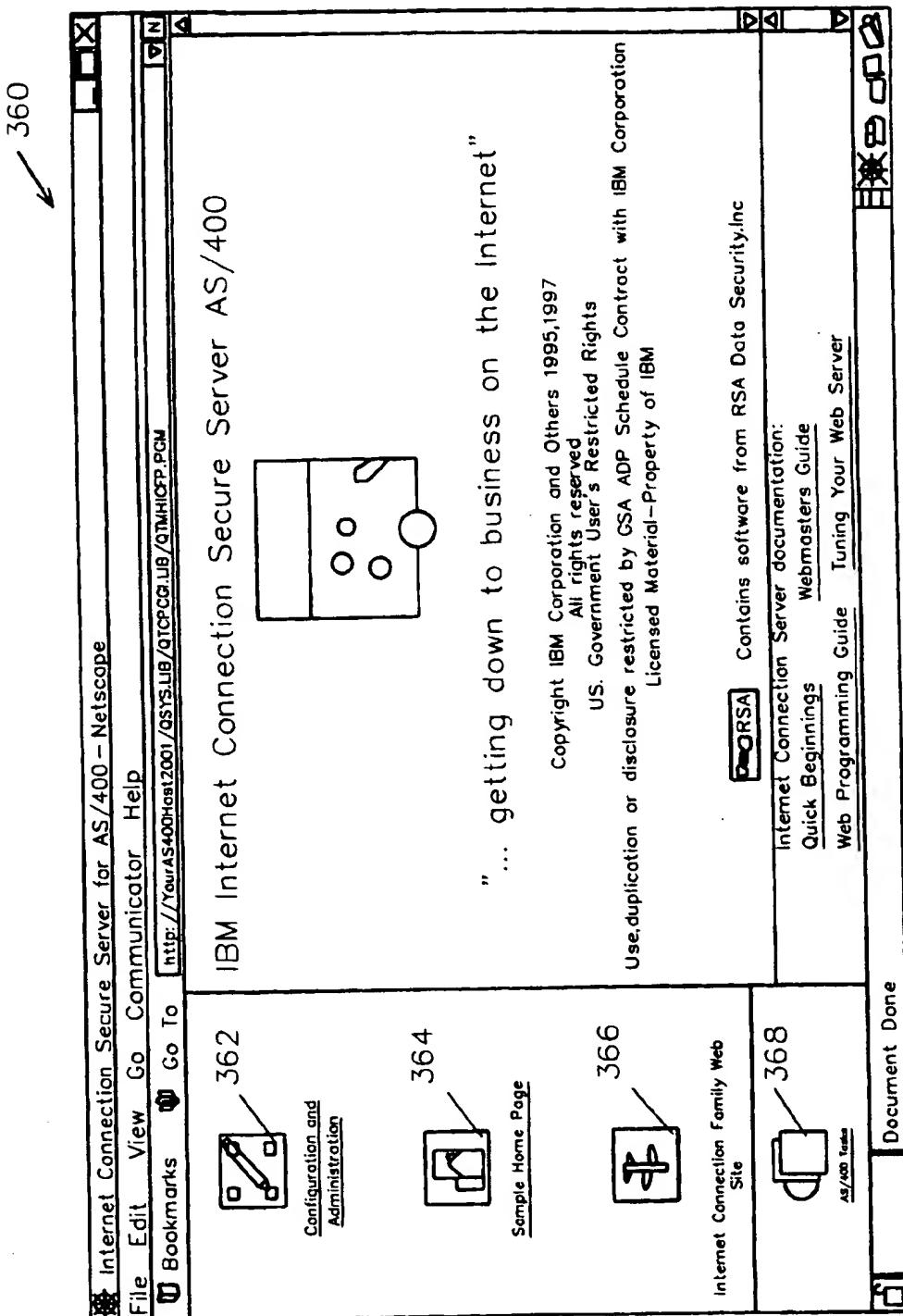


FIG. 8

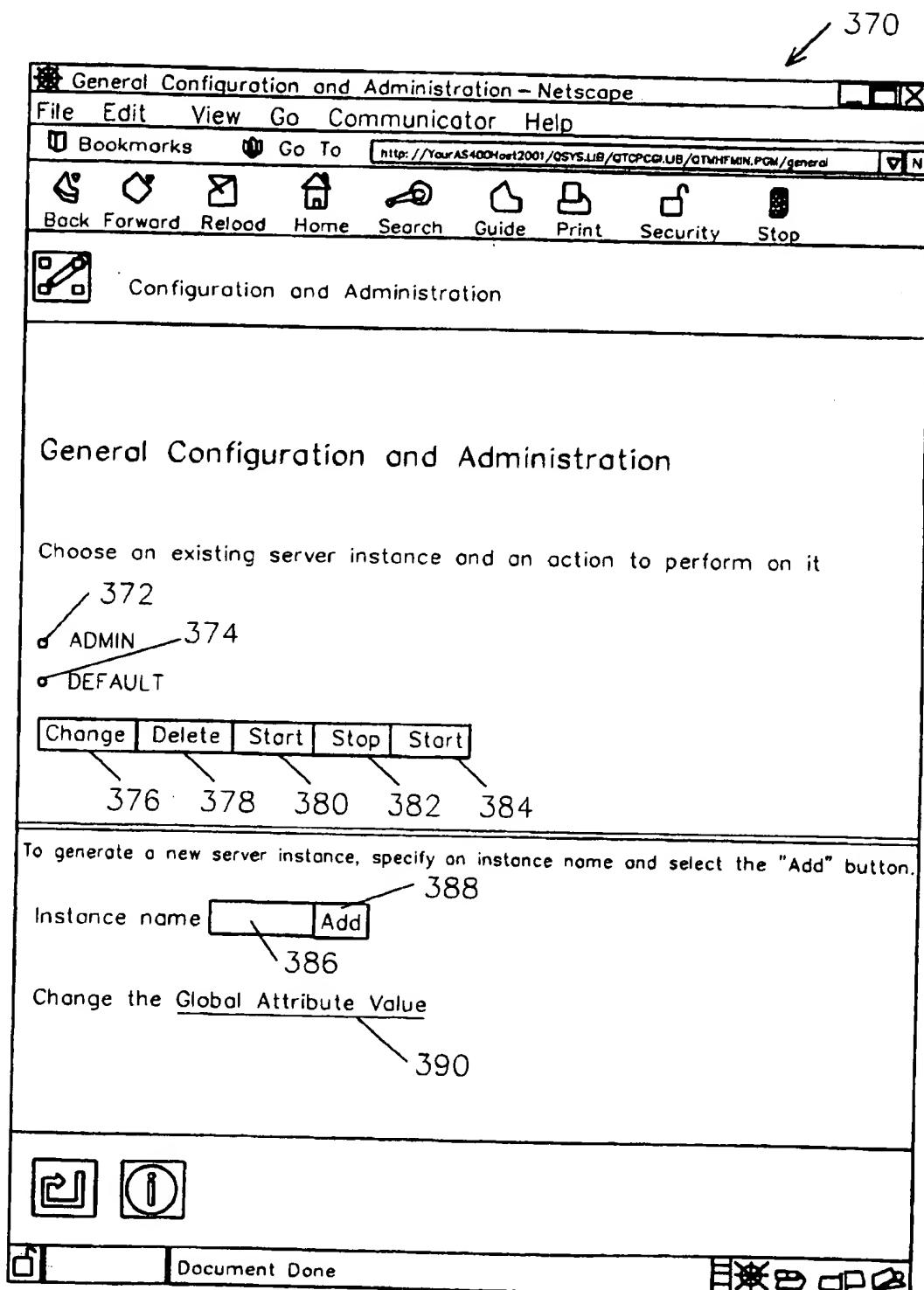


FIG. 9

10/19

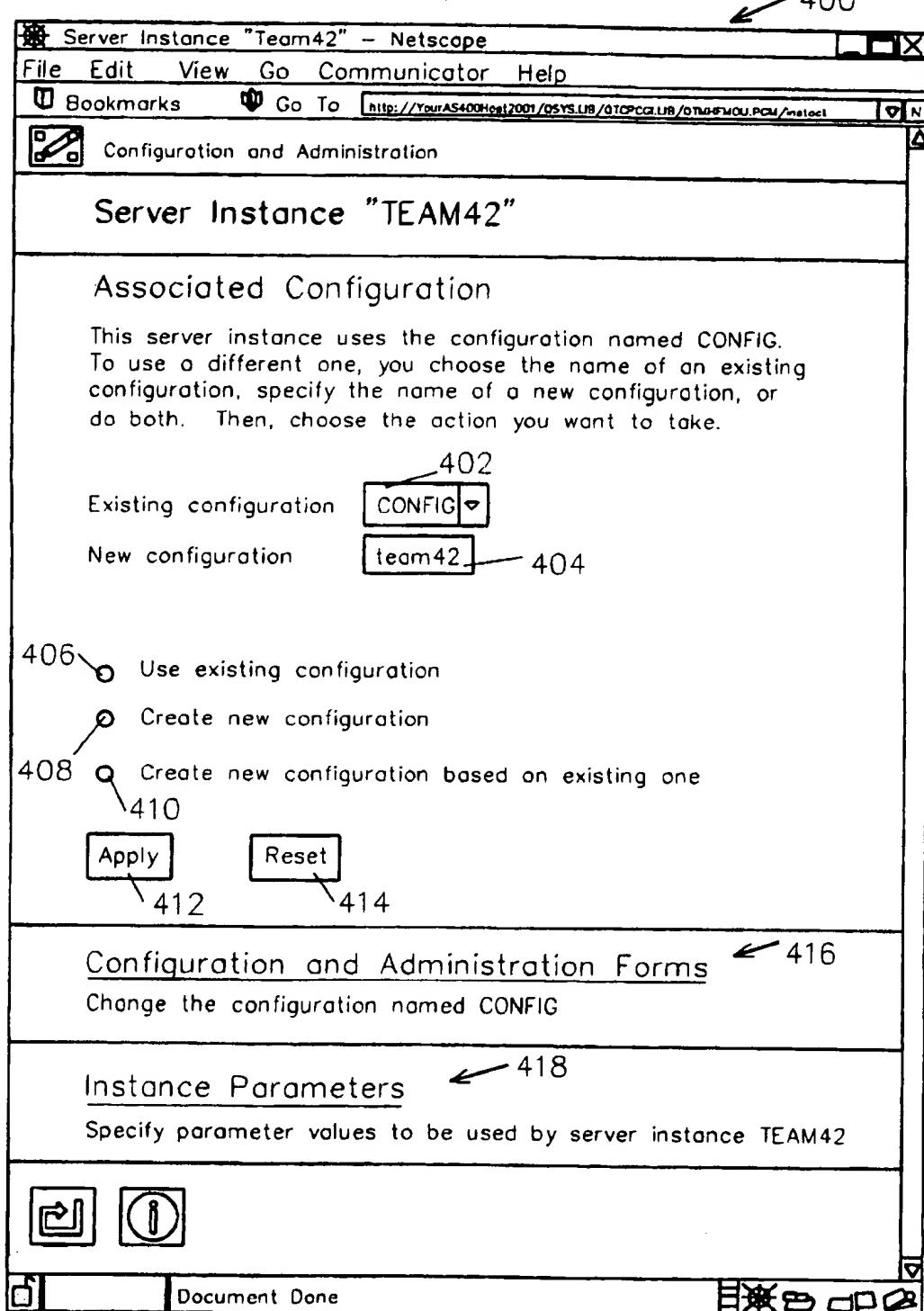


FIG. 10

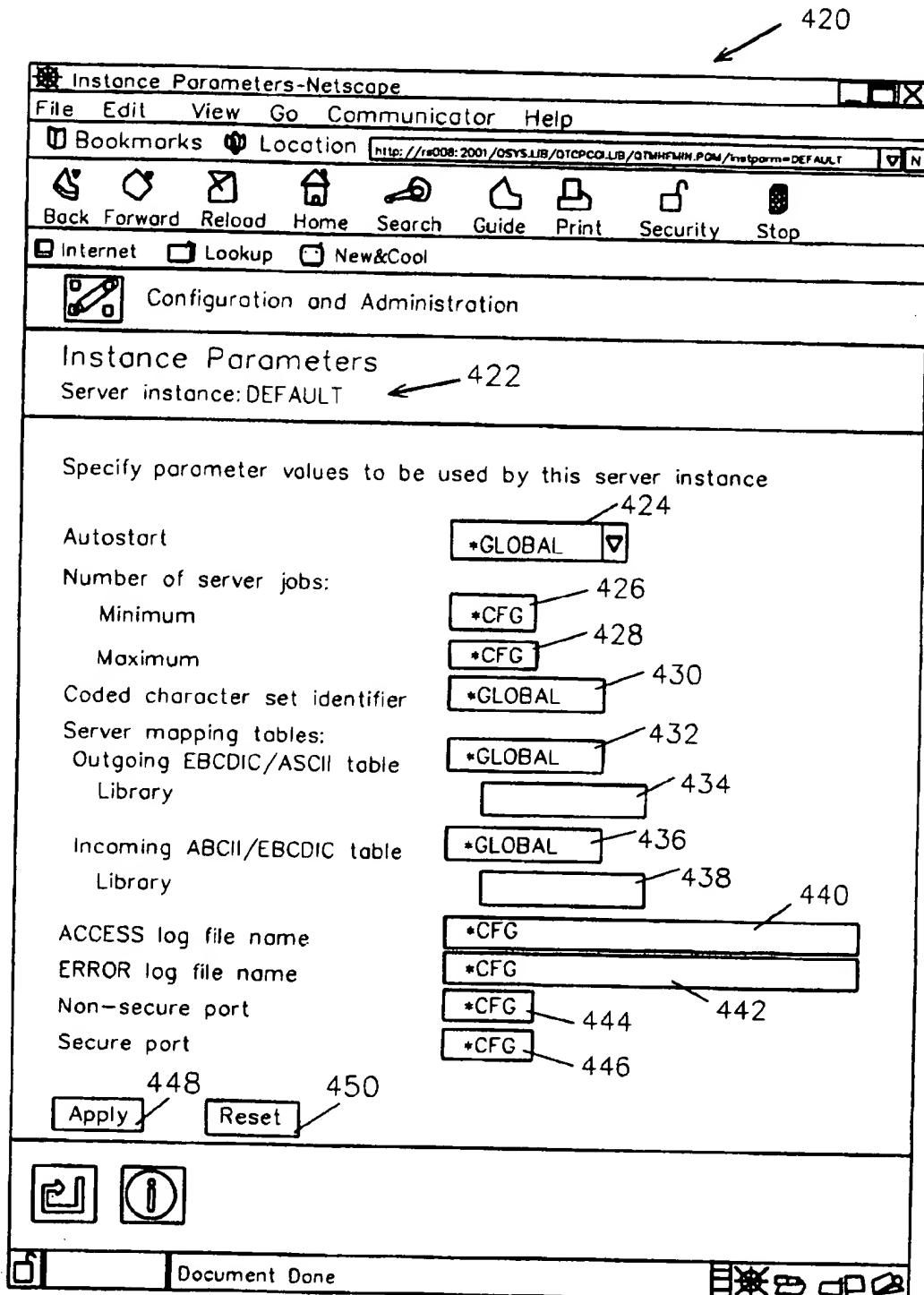


FIG. 11

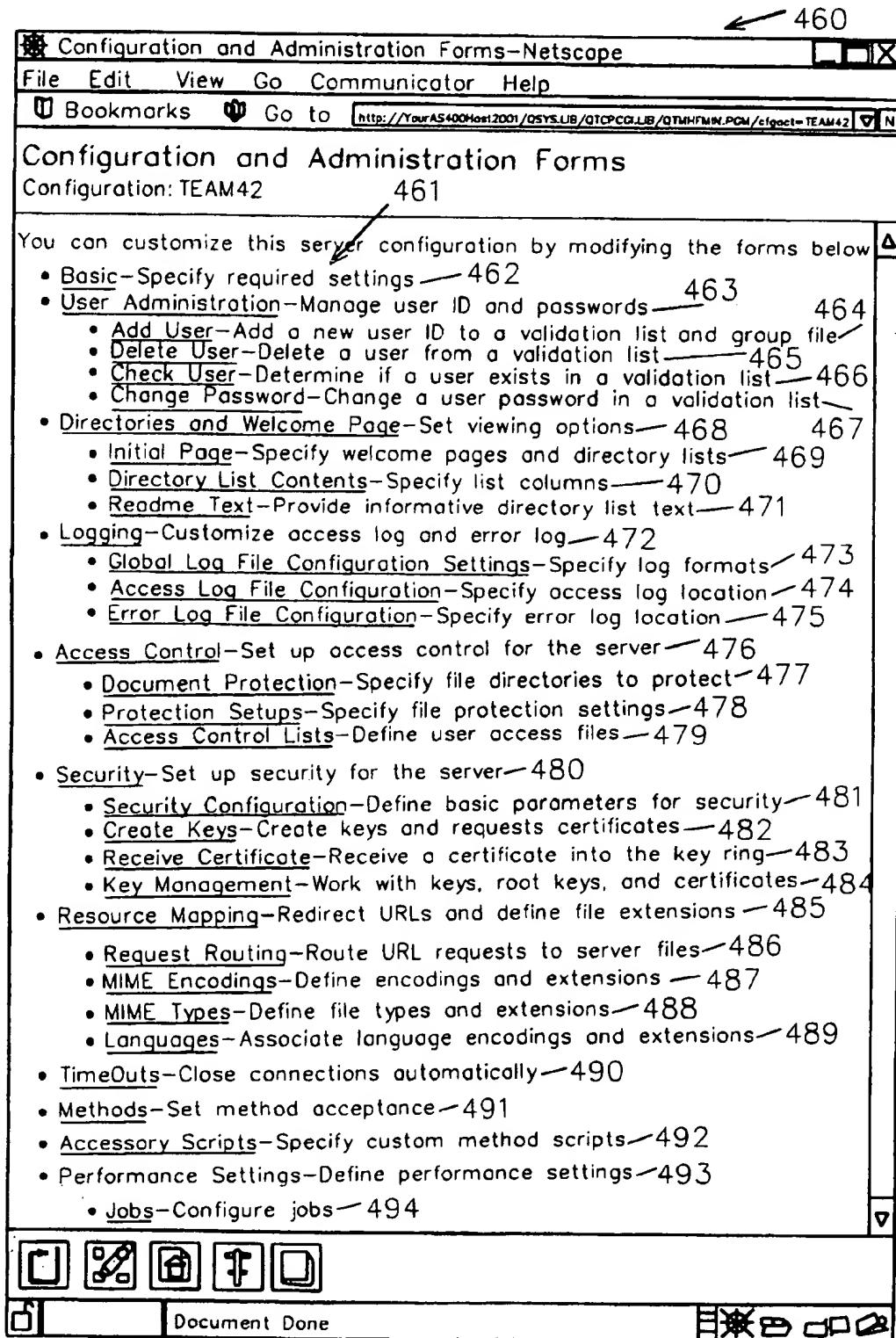


FIG. 12

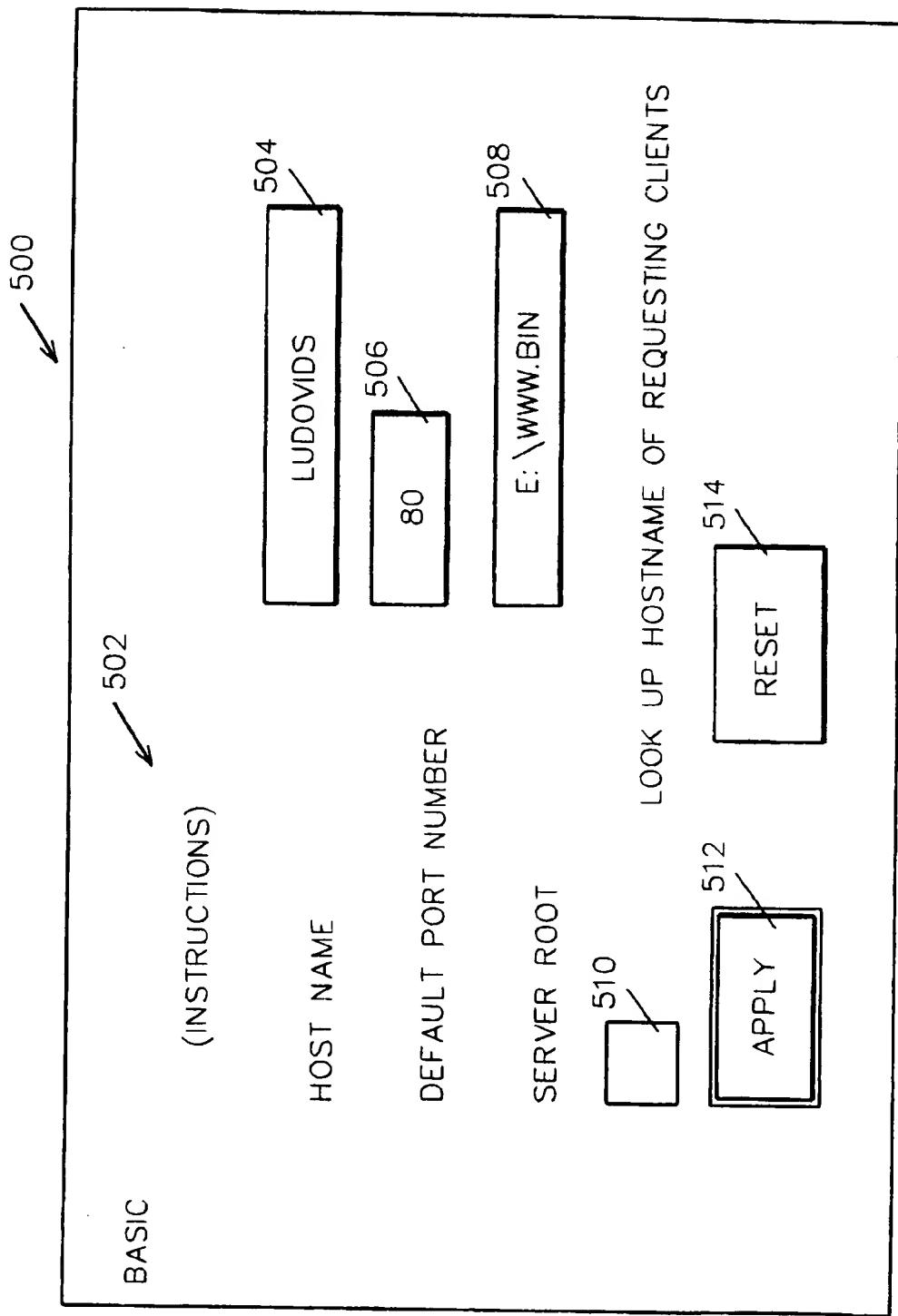


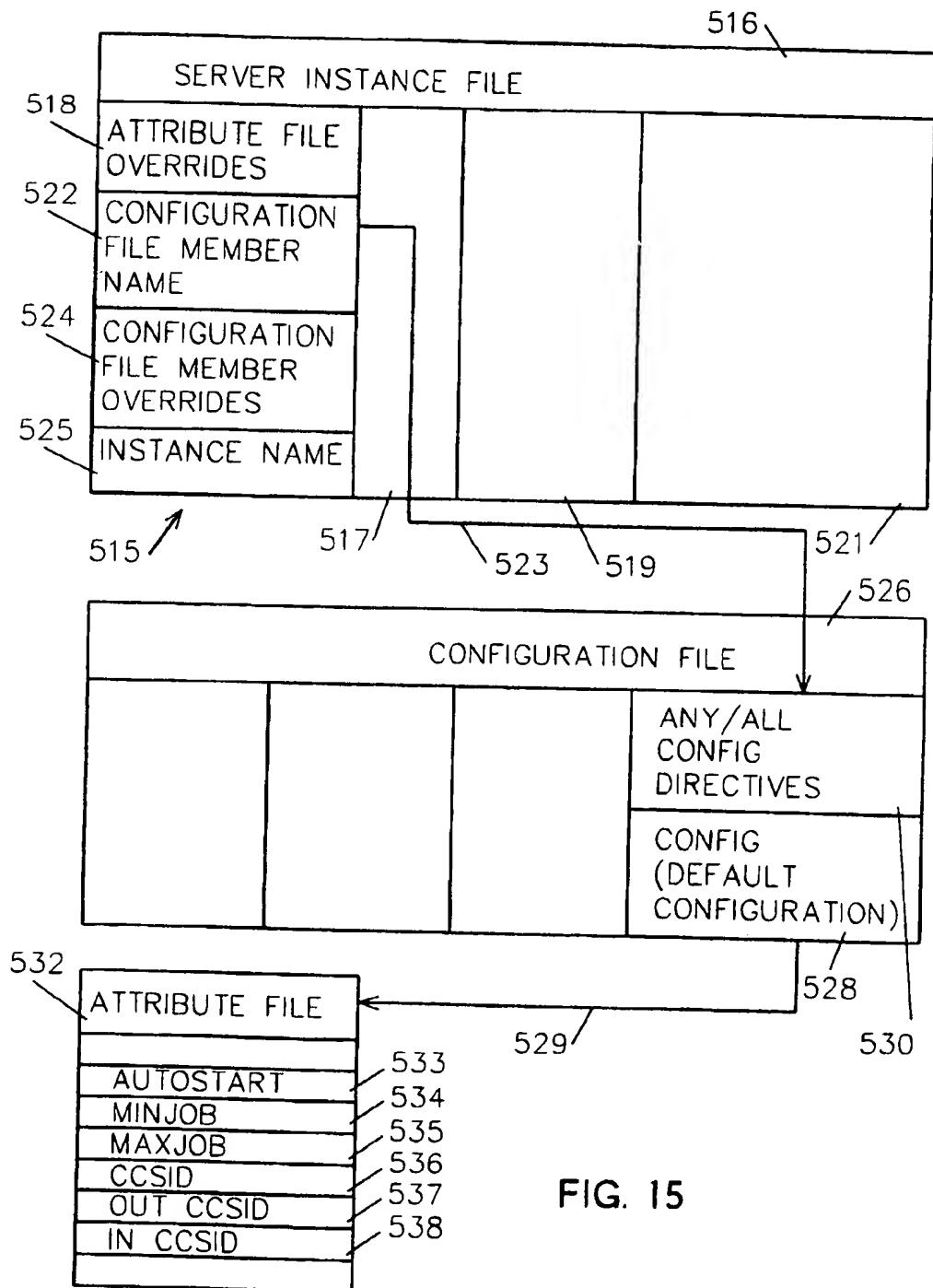
FIG. 13

FIG. 14

520  
↙

## CONFIRMATION

DIRECTIVE:	HOSTNAME	LUDOVIDS
DIRECTIVE:	PORT	80
DIRECTIVE:	SERVERROOT	E:\WWW.BIN
DIRECTIVE:	DNS-LOOKUP	OFF
DIRECTIVE:	IMBEDS	OFF
< MESSAGE >		



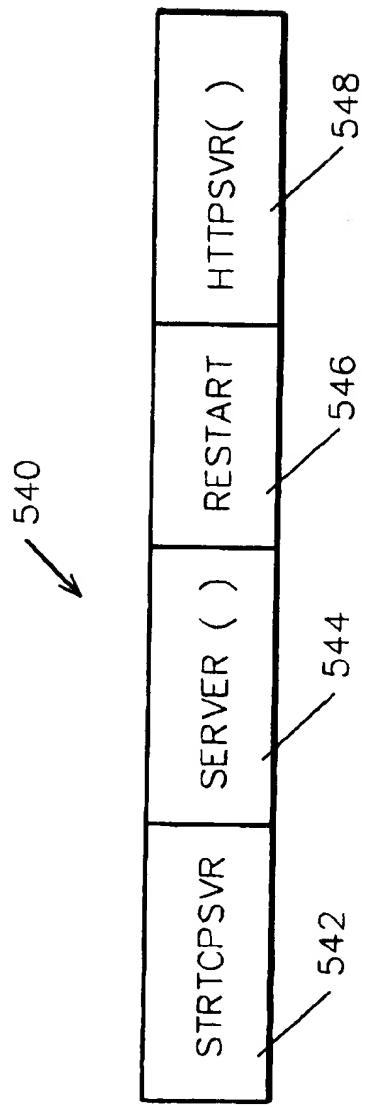


FIG. 16

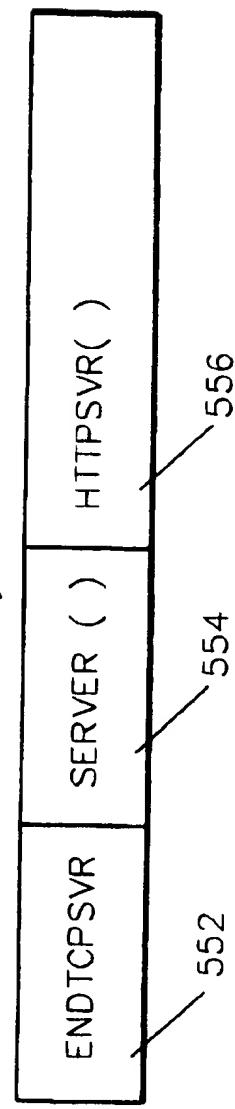


FIG. 17

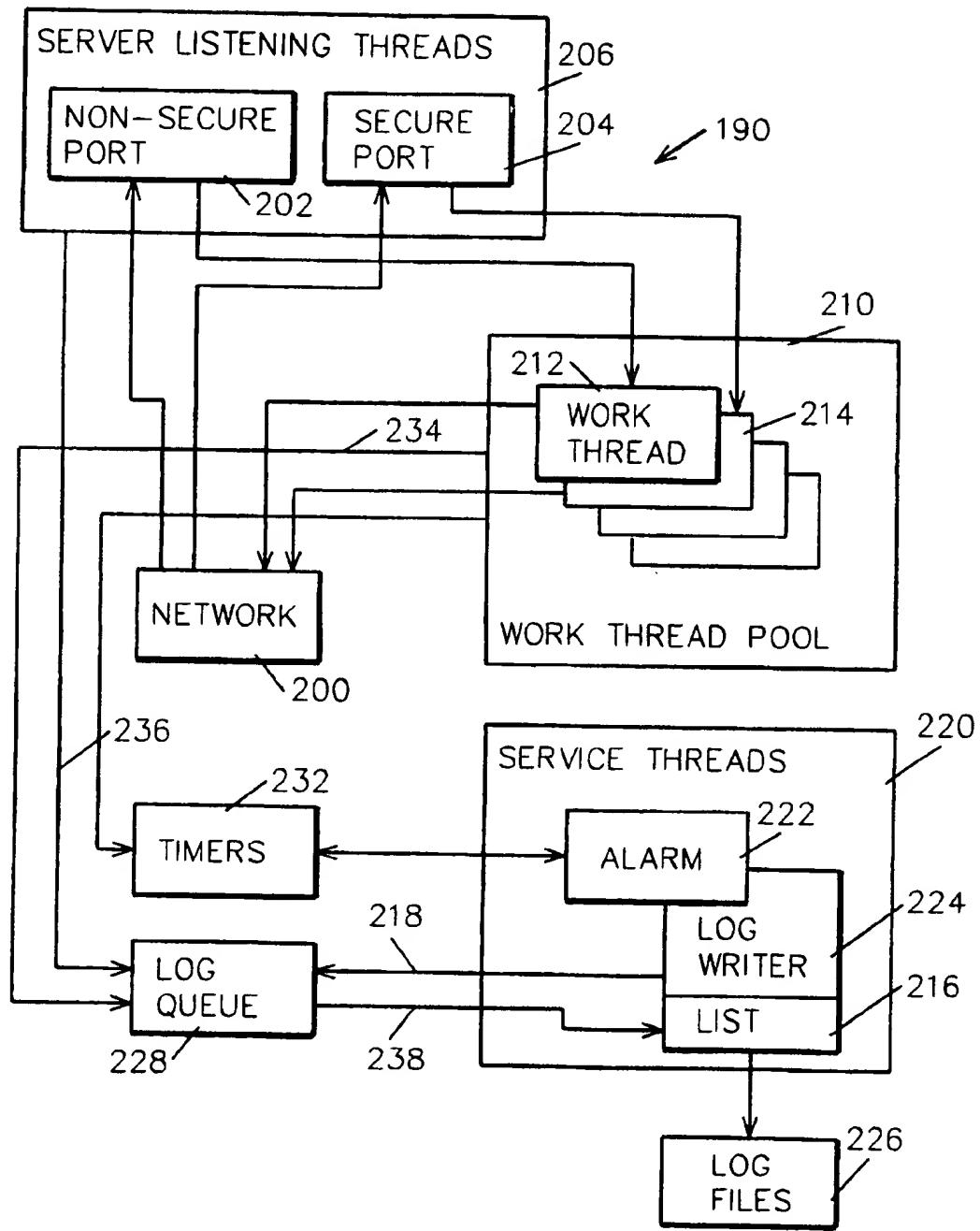


FIG. 18

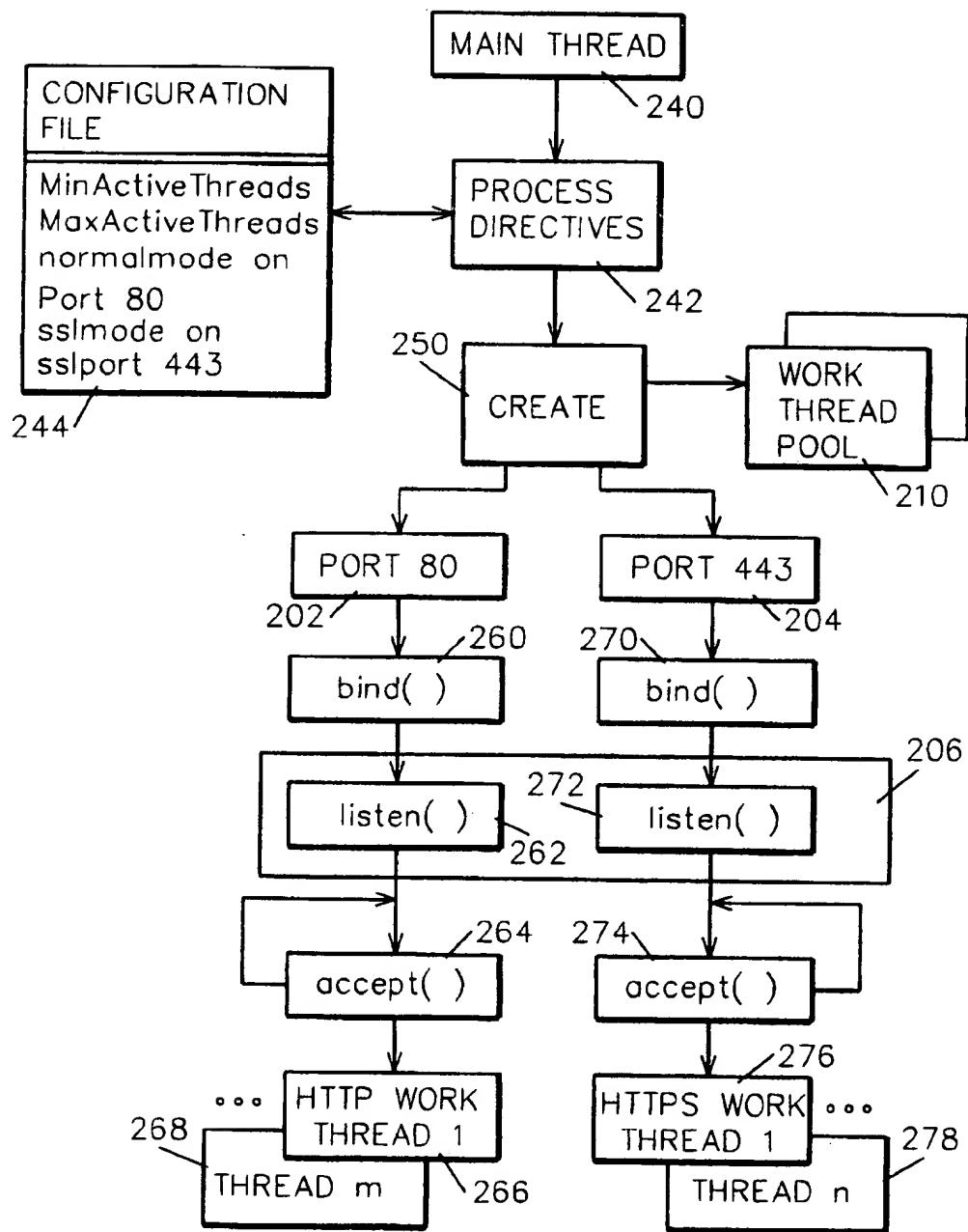


FIG. 19

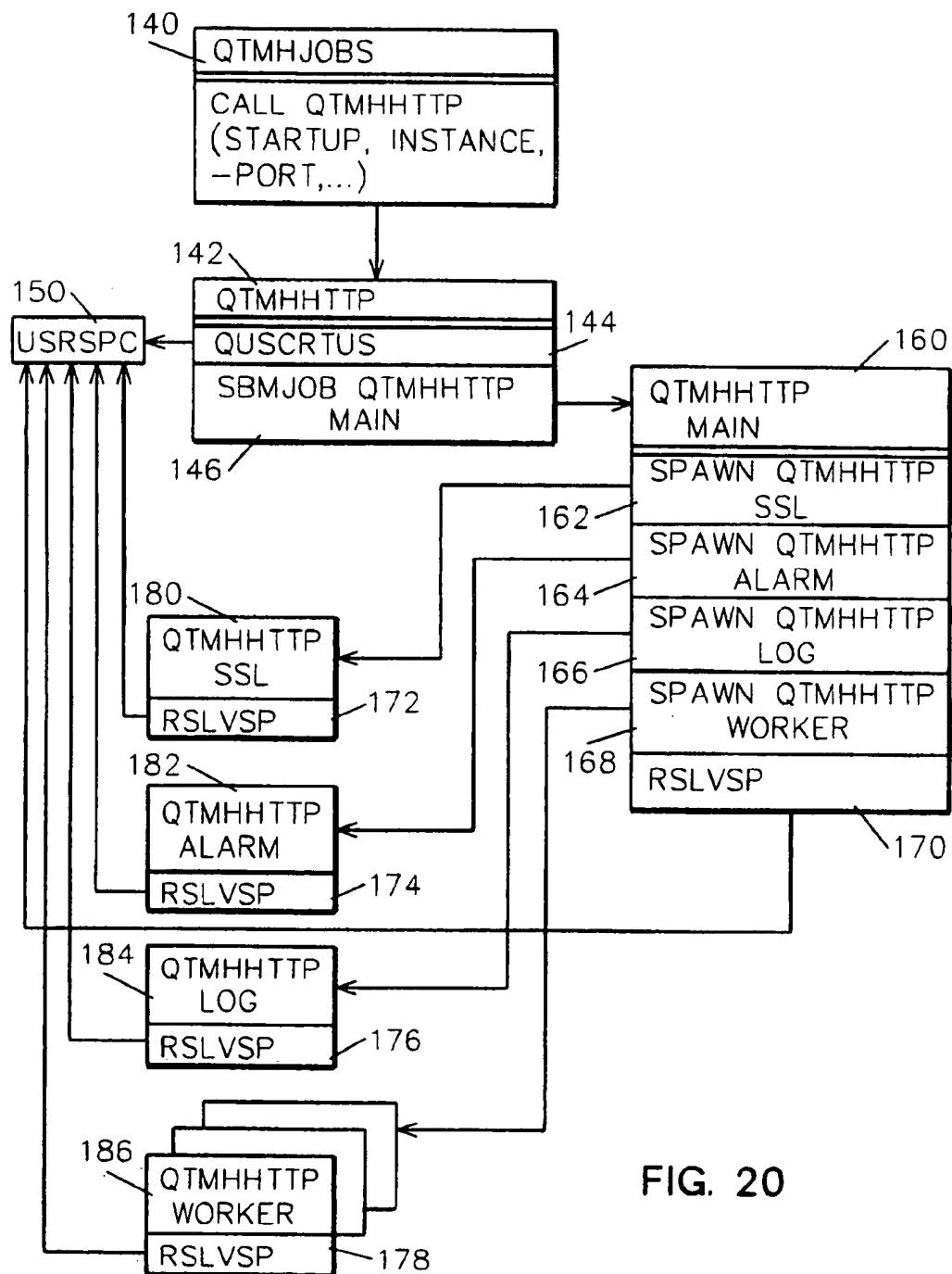


FIG. 20

## SYSTEM AND METHOD FOR AN ADMINISTRATION SERVER

### CROSS-REFERENCES TO RELATED APPLICATIONS

U.S. patent application Ser. No. 09/135,149 filed Aug.17, 1998, entitled "SYSTEM AND METHOD FOR CONFIGURING AND ADMINISTERING MULTIPLE INSTANCES OF WEB SERVERS", and Ser. No. 09/135,253 filed Aug.17, 1998, now U.S. Pat. No. 6,272,518 entitled "SYSTEM AND METHOD FOR PORTING A MULTITHREADED PROGRAM TO A JOB MODEL", and filed concurrently herewith and assigned to the same assignee. They contain related subject matter incorporated herein by this reference.

### BACKGROUND OF THE INVENTION

#### 1. Technical Field of the Invention

This invention pertains to configuration and administration of system components using World Wide Web (WWW) technology. In particular, this invention provides a plurality of instances of web server, with one of them being preconfigured for this administration function to allow web browsers to configure web-configurable components.

#### 2. Background Art

The IBM AS/400 system traditionally uses 5250 terminal protocol to configure and administer system components, such as start or stop a server, change functional attributes, authorize users, and so forth. This 5250 terminal protocol requires a 5250 terminal or 5250 emulator that is known as a green screen display. A graphical user interface (GUI) presents to a user a much more user-friendly interface than a green screen display, and there is a need in the art to provide a graphical user interface for system configuration and administration functions.

It is quite common today for any system to have a web server that hosts some site or sites. Such sites provide web content that is made available to web browsers. In general the goals of an enterprise with such a site are typical business goals that pertain to return on investment or customer satisfaction. Typically, such an enterprise has system sites that deal with business applications (such as an electronic shopping mall), or information sites (such as the company's product specifications, or employee benefit plans). These sites share in common the protocols that make web browsing possible. These include use of the HyperText Transfer Protocol (HTTP) and HyperText Markup Language (HTML), a Common Gateway Interface (CGI) that allows system administrators to write dynamic web applications, and the content itself, that is the HTML files, images, java applets, wave (audio) files, or other multimedia resources known to (that is, accessible by) browsers.

Some servers have the ability to manage other servers. However, systems today do not have multiple copies of those servers, and there is a need in the art for an administration server capable of managing multiple copies or instances of servers.

Further, there is a need in the art for an administration Internet connection server (ICS) that is a web (HTTP) server for serving a specialized set of applications that provide for configuration and administration of web enabled system components. These system components can be anything on an enterprise or site system that have an application written for and served by an administration server for the purpose of being configured by a web browser.

However, for security purposes, this administration Internet connection server must not share commonality with other ICS HTTP server(s) serving the web content that is made available by the site system to web browsers. Such an administration server preferably allows a user to configure certain aspects of the server, such as access and error logging, or the ability to run secure HTTP transactions (also referred to as HTTPS). However, this administration server must be controlled with respect to serving its content to browsers irrespective of such configuration by the user.

Further, there exists the need in the art to provide an administration Internet connection server (also referred to as an administration server) which is isolated from and therefore does not detract in any way from other, or "normal", HTTP server(s) at the site, and does not impose any requirements that the system use its "normal" web server(s) for what is considered system administration and configuration. It is particularly important that any changes the site makes to configuration of its "normal" HTTP server(s) (good or bad) cannot affect the configuration of the administration server. Furthermore, it is a requirement that any heavily loaded "normal" web server(s) at a site will not affect performance of the administration server, and conversely, the act of using the administration server, that is configuring and administering some system component, will not have any effect on the performance of other HTTP server(s) at the site.

It is an object of the invention to provide an improved administration server.

It is a further object of the invention to provide an administration server which serves administration and configuration applications to a browser's graphical user interface.

It is a further object of the invention to provide an administration server which serves to a browser capability for administering and configuring web enabled system components.

It is a further object of the invention to provide an improved administration server which is isolated from other servers at the site.

It is a further object of the invention to provide an improved administration server which is isolated from other servers at the site such that loading of either does not adversely impact the other.

It is a further object of the invention to provide an improved administration server which enables the management of multiple copies or instances of servers.

It is a further object of the invention to provide an administration server implemented as a browser.

### SUMMARY OF THE INVENTION

In accordance with the invention, a system and a method are provided for serving HTML pages to web browsers for the purpose of administration and configuration. A plurality of instances of WWW servers is provided, with one such instance including a configuration file which is restricted in usage and not alterable by way of any HTML configuration or administration forms.

Other features and advantages of this invention will become apparent from the following detailed description of the presently preferred embodiment of the invention, taken in conjunction with the accompanying drawings.

### BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a high level system diagram of the administration server of the invention.

FIG. 2 is a high level system diagram illustrating the administration server of the invention as one instance of multiple server instances.

FIG. 3 is a diagrammatic representation of the configuration and server instance forms presented by the administration server of the invention at a graphical user interface.

FIG. 4 is a high level system diagram illustrating the main components of the system of the invention.

FIG. 5 is a flow diagram of the method of the invention.

FIG. 6 is a flow diagram of the \*ADMIN site of the preferred embodiment of the invention.

FIG. 7 is a diagram illustrating the AS/400 tasks page.

FIG. 8 is a diagram illustrating the front page, or first page for configuration and administration of ICS in accordance with a preferred embodiment of the invention.

FIG. 9 is a diagram illustrating the general configuration and administration page for adding a new ICS server instance or to manage an existing one in accordance with the preferred embodiment of the invention.

FIG. 10 is a diagram illustrating a server instance page in accordance with the preferred embodiment of the invention.

FIG. 11 is a diagram illustrating a representation of instance parameters in accordance with the preferred embodiment of the invention.

FIG. 12 is a diagram illustrating a representation of configuration and administration in accordance with the preferred embodiment of the invention.

FIG. 13 is a diagram illustrating a basic page.

FIG. 14 is a diagram illustrating a confirmation page.

FIG. 15 is a diagram illustrating secure server instance management files and formats.

FIG. 16 is a diagram illustrating the format of the start server command.

FIG. 17 is a diagram illustrating the format of the end server command.

FIG. 18 is a high level system diagram illustrating server threading.

FIG. 19 is a high level flow diagram illustrating server threading flow.

FIG. 20 illustrates the job structure of the HTTP server of the preferred embodiment of the invention.

#### BEST MODE FOR CARRYING OUT THE INVENTION

In accordance the preferred embodiment of the invention, an hypertext transfer protocol (HTTP) web server is provided for serving browsers. A web server that supports the industry standard HTTP protocol is given a hardened (e.g., read only) configuration. A task page is provided that provides links to any application that can be configured via a browser. This one task page is served by this administration HTTP server. Upon selecting a link to a configurable application, the selected pages which are also served by the administration HTTP server, are used to complete the configuration task. The user interface throughout this process is the browser.

The pages, and forms contained therein, are accessed by a series of links from the tasks page, and provide the graphical user interface (GUI) for configuring and administering those applications. Examples of such applications include internet connection secure server, firewall, net.data, digital certificate manager, net.commerce and net.question. These become a link off of the tasks page and each provides

a series of web pages which are traversed to achieve the desired configuration.

In accordance with the invention, forms are designed and implemented so that an internet connection server can be configured. A configuration file for an Internet connection server is built by selecting and entering values upon these forms. One or more copies or instances of the Internet connection server are provided, and these forms are used to manage these multiple instances. The administration server of this invention enables creation, selection and modification of other server instances by leading an authorized user at a browser through a series of GUI forms to accomplish the change, addition, start, stop, or other action.

For example, a plurality of instances of Internet connection servers (ICSs) exist at a site and need to be managed. A form is presented by the administration server listing this plurality of other servers, and the user selects the instance (that is, the other server) he needs to administer (start, stop, change, add, delete). Upon selecting a server and an action, a page is presented or displayed at the browser by the administration server which will allow further configuration, the content of that page depending upon the action selected.

In accordance with the preferred embodiment of the invention, there is provided an administration server that is accessed via a web browser.

Referring to FIGS. 1 and 2, ADMIN server 310 represents an instance of an HTTP server. In this preferred embodiment, there are provided an instance file 318 with one member and a configuration file with one member pair 314, 316 called ADMIN for installation separate and distinct from another instance 311 and configuration file 317 that are also provided for installation, but for a system administrator's use. A global attributes file 325 is provided. Instance file 317 overrides the contents of the config file(s) 314, 316, which overrides the contents of the attributes file 325. Attributes file 325 is referred to as the global attributes file because there is only one file member, used by all server instances; whereas each instance uses a unique instance file member 318 and a particular configuration (config) file member 314 (except admin server 310, which uses 2 config file members 314, 316).

In accordance with a preferred embodiment of the invention, an Internet connection secure server has a default setup that includes two HTTP server instances 310, 311. One instance is the ADMIN server 310 that is required for configuration and the other is DEFAULT server 311. Default server 311 can be started as-is with all of its default settings 317, 319, 321, 323. The server administrator can customize the default server 311 for his own environment and add additional servers as desired.

Referring to FIG. 2 in connection with FIG. 7, in accordance with a preferred embodiment, ADMIN server 310 serves an AS/400 configuration home page 350 that links to other browser configurable products, such as is represented by link 352. In operation, when the user (that is, the system administrator) points his browser (this is analogous to saying <http://as400hostname.domain:2001>) to the system administration home page, the browser is presented with a page 350 asking what is to be administered or configured, including a hot link 352 to HTTP server configuration pages, or alternatively a different page showing configuration screens for the other products.

Upon selecting link 352, the user is presented initial page 360. Upon selecting configure HTTP servers 362 on initial page 360, the system administrator is presented a general configuration and administration form 370 (also referred to

as the server instance form or instance manager page). The system administrator is in this manner allowed to select either an instance 371 to work with, add a new instance, or change the default attributes that potentially affect all HTTP servers. Additionally, a user may choose the other hotlinks 364, 366, 368 on this initial page 360 to see some sample home pages, some URLs for help, or to return to initial page 350.

Referring to FIG. 3, instance manager page 370 provides various choices for working with an instance 371, including CHANGE 376, DELETE 378, START 380, STOP 382 and RESTART 384. Appropriate messages for these commands are shown after the command has executed. Help text is provided to guide a user through the steps. These selections are then turned into appropriate control commands in the case of START 380, STOP 382 or RESTART 384, or will delete an instance 371 in the instance file 318 if DELETE 378 is selected.

In accordance with a preferred embodiment of the invention, an interface for configuring the AS/400 Secure Web Server includes web pages and CGI scripts and a configuration file validation program. However, the web-based configuration function of the preferred embodiment of the invention does not use a separate validation program. Rather, validation of user entries in the configuration pages 316 is built into the CGI programs. Which CGI program gets invoked depends on which button (Apply, Change, . . .) is pushed or which link (i.e., Global Attribute Values) is clicked, regardless of the rest of the page content. These CGI programs produce the code set forth in Tables 4-9.

The CGI scripts read current settings from the configuration file and build configuration pages filled in with those settings. CGI scripts are also used to read the values contained in the configuration pages and write those values out to the configuration file. These scripts may be modified to read or write from or to a configuration file 314 or an instance file 318, as appropriate. As will be apparent to those skilled in the art, this is common cgi usage, and need not be further described.

In accordance with a preferred embodiment of the invention, secure server configuration proceeds as follows:

1. The user opens the Internet Connection Server for AS/400" page.

2. As each general configuration and administration form 370 (FIG. 3, a specific example of which is illustrated in FIG. 9 as form 370) is accessed, the "in" CGI script 391 obtains the appropriate values from the configuration file 316, instance file 318, global attributes file 325 and builds, for example, a page 400 (FIG. 10), which is filled in with those values and displayed to the user at the browser.
3. If the user clicks on the "Apply" button 412 (FIG. 10), an "out" CGI script 391 calls a validation routine. If valid, the configuration directives are written to the instance file 318 and a confirmation page is returned. Otherwise, an error page is returned.
4. If the user clicks on "Reset" button 414 (FIG. 10), the "in" CGI script rebuilds the current page. The web browser 304 resets the input fields.

Referring to FIG. 4, another view of the main components characterizing the environment of the administration and configuration system \*ADMIN 310 of the invention includes system 300 and web browser 304 in communication over a network 302. In accordance with the preferred embodiment of the invention, web browser 304, any industry compliant web browser, is the user interface required to use this \*ADMIN server 310. (An alternative user interface is described hereafter in connection with start TCP server

command 540 and end TCP server command 550, FIGS. 16 and 17, respectively.) Web browser 304 communicates through network 302 using HTTP protocols to the Administration (HTTP) Internet Connection Server (ICS) 310, also referred to as \*ADMIN server 310.

\*ADMIN server 310 is started, and it's functionality is controlled, via administration instance file 318, read-only administration configuration file 314, read-write administration configuration file 316, and global attributes file 325. The system administrator can change configuration file 316 in order to maintain system compatibility; however, read-only administration configuration file 314 contains those parameters which are required to be served by \*ADMIN server 310 to browser 304, and cannot be altered by the system administrator. That is, read-only configuration file 314 is hardened: while all directives can be put in read-write configuration file 316, they will not override all of the directives that are in read-only file 314. The contents of file 314 are shown in Table 3. Server 310 provides a Common Gateway Interface (CGI) 308 for the purpose of executing CGI applications 306. These applications 306 are described later.

It is quite common today for any system to have a web server that hosts some site or sites. Such sites provide web content 312 that is made available to web browsers 304, such as information sites (for example, the company's product specifications, or employee benefit plans). These sites share in common the protocols that make web browsing possible, that is the HyperText Transfer Protocol (HTTP) and HyperText Markup Language (HTML), a Common Gateway Interface (CGI) 308 that allows system administrators to write dynamic web applications 306, and content files 312 such as HTML files, images, java applets, wave (audio) files, or other multimedia resources known to browsers.

Administration (HTTP) Internet Connection Server (\*ADMIN server) 310 is a web (HTTP) server, however it serves a specialized set of applications 306, namely those that deal with the configuration and administration of web enabled system components. These system components 306 can be anything on the system that has been web enabled. By web enabled is meant that an application 306 has been written and served via the \*ADMIN server 310 for the purpose of being configured via a web browser 304.

\*ADMIN server 310 is unique (from other servers) in that it does not share any commonality with the System administrator's ICS (HTTP) server(s) 311—that is it is a separate instance of the ICS server, with separate configuration and instance files 316 and 318, and a specific URL (Uniform Resource Locator). \*ADMIN server 310 also has a special "read-only" configuration file 314 that is shipped and installed with the \*ADMIN server 310 so the operation of the \*ADMIN server 310 can be controlled with regards to serving its content, irrespective of what the system administrator may want to configure from browser 304. In accordance with this preferred embodiment, the system administrator is allowed to configure certain aspects of the server such as access and error logging, or the ability to run secure HTTP (HTTPs) transactions.

Read-only configuration file 314 is read by the web server before read-write configuration file 316. There are some directives which affect web serving in such a way that the first directive encountered that covers a particular web activity is the directive that will be used to control that activity. In effect, these directives cannot be overridden. They include Protect, Map, Pass, and Exec. Thus, much of the usage of the server that is controlled by the contents of

the read-only file 314 will not be overridden by the contents of the read-write file 316. Instance parameters 318 would override the contents of the read-only file 314, but are limited in number, and are shown in FIG. 11. For example, the contents of the read-only file 314 allows the tasks page to be served, and this behavior cannot be overridden.

Thus, \*ADMIN server 310 does not detract in any way from the system administrator's "normal" HTTP server(s) 311, and does not impose any requirements that the system administrator use his normal web server(s) 311 for what is considered system administration and configuration. The web-site (content) the system administrator wants to host is separate and distinct from \*ADMIN server 310, and any changes the system administrator at browser 304 makes to his "normal" HTTP server(s) 311 configurations 317 (good or bad) cannot affect the configuration 314, 316 of \*ADMIN servers 310. Furthermore, any heavily loaded system administrator's "normal" web server(s) 311 will not affect performance of \*ADMIN server 310, and conversely, the act of using \*ADMIN server 310, that is configuring and administering some system component 306, will not have any effect on the performance of the system administrator's other HTTP server(s) 311.

In order to keep the content of \*ADMIN server 310 separate and distinct from the "normal" HTTP server(s) 311, it is necessary to provide a unique URL (Uniform Resource Locator) so browser 304 can access the home page of \*ADMIN server 310. There are two different approaches to achieving this unique URL. The first is to provide a unique host address, and the URL would be: `http://some_unique_address`. However, the system administrator would have to configure this, thereby implying a multi-homed host 300, and obtain a separate IP address which may prove to be difficult. The second approach is to "port qualify" the URL so that instead of using the well known port for HTTP (which is port 80), and the well known port for HTTPS (which is port 443), a different port 303, 305 is chosen for each protocol, namely port 2001 for HTTP, and 2010 for HTTPS. Since it is possible to have a conflict with other socket applications, these ports are configurable. The URL for \*ADMIN server 310 then becomes `http://your_host_name:2001`—this is what the system administrator would enter at browser 304 to access the home page 350 (FIG. 7) of \*ADMIN server 310.

In accordance with this preferred embodiment, implementation of \*ADMIN server 310 includes the following: an industry standard HTTP server 310 that has a CGI interface 308; the ability to have (execute) multiple copies of an HTTP-ICS server, including \*ADMIN server 310 and at least one other server 311; control of this \*ADMIN server 310 by means of a configuration file 314, 316, instance file 318 and global attributes file 325; separate and distinct configuration and instance files 317 for the other ICS server(s) 311; a "read-only" configuration file 314 that guarantees operation of the \*ADMIN server 310 (this file 314 is shipped by the manufacturer to the customer whose system 301 this is); a "read-write" configuration file 316 giving the system administrator the ability to configure some aspects of \*ADMIN server 310; the ability to configure \*ADMIN server 310 to bind to ports 303, 305 different than the well known HTTP and HTTPS ports. CGI interface 308 provides connection between the configuration pages and the CGIs.

In operation, referring to FIG. 5, after a start command 320 is received, in step 322 \*ADMIN instance file 318 is read and appropriate parameters are passed to the main thread of the program. In step 324, the main thread then

reads the configuration files 314, 316, builds a rules list from them, and then connects to the HTTP (and perhaps HTTPS) ports. In steps 326 and 328, the main thread of server 310 listens (waits) on the HTTP port, and the SSL listener thread 5 listens (waits) on the HTTPS port for work to arrive (that is, requests from browser 304), and then passes these requests to a non-busy worker thread 330 from a managed pool of worker threads. Worker thread 330 is so named since it does all the work, reading, parsing and comparing the request against the rules list, and then acting on the rules that are encountered. When processing by worker thread 330 is complete, it returns information of some kind to browser 304, which may be the resource 312 requested, output of a CGI application 306, or an error message.

15 The threads model used for executing ADMIN server 310 is further described in copending U.S. patent applications Ser. No. 09/135,149, filed Aug. 17, 1998, entitled "SYSTEM AND METHOD FOR CONFIGURING AND ADMINISTERING MULTIPLE INSTANCES OF WEB SERVERS", and 09/135,253, filed Aug. 17, 1998, now U.S. Pat. 6,212, 20 518, entitled "SYSTEM AND METHOD FOR PORTING A MULTITHREADED PROGRAM TO A JOB MODEL", the teachings of which are herein incorporated by this reference.

25 In Table 1, a further elaboration on the operation of of \*ADMIN server 310 is set forth in pseudo code. In Table 1, comment lines are preceded by //.

TABLE 1

30	*ADMIN Server
	// pseudo code for Server 310
	// error handling is not shown here but is required for
	// product level implementations
35	START TCP SERVER COMMAND 320 RECEIVED INDICATING START *ADMIN SERVER 310 OR START TCP COMMAND RECEIVED AND AUTOSTART="YES" READ AND PARSE *ADMIN SERVER INSTANCE FILE 318 START MAIN *ADMIN THREAD WITH ANY OVERRIDE PARAMETERS
40	//main thread processing SETUP SIGNAL HANDLERS // useful for such things as realizing the end top server // command has been issued. This way polling is avoided // for the end command, and the main loop can do real work READ AND PARSE CONFIGURATION FILES 314, 316 BUILD RULES LIST // this rules list is what governs the behavior of the
45	// server FOR ALL SERVER ATTRIBUTES IN CONFIGURATION FILE 314, 316 // server attributes include such things as number of worker // threads code pages for translations, access and error // logging if turned on
50	// additional listener thread 328 if running HTTPS also SETUP APPROPRIATE ERROR, LOGGING, ALARM THREADS SETUP WORKER THREADS 330 // if all goes well, bind to specified socket and wait // for work // usually 2001 for *ADMIN HTTP // usually 2010 for *ADMIN HTTPS (secure HTTP transactions)
55	CONNECT TO SOCKET SPECIFIED IN CONFIG OR OVERRIDE PARMS OR SERVICES TABLE WAIT IN ACCEPT LOOP FOR BROWSER REQUESTS WHEN REQUEST ARRIVES, PASS THE SOCKET DESCRIPTOR TO A WORKER THREAD 330 BACK TO WAIT IN ACCEPT LOOP
60	// a worker thread that is not busy is selected from the // pool 330 of worker threads. // now worker thread has the socket descriptor READ HTTP REQUEST FROM SOCKET APPLY RULES LIST TO HTTP REQUEST IF A MATCH OCCURS
65	// there may be more than one, but first match wins PROCESS RULE WITH "THIS" HTTP REQUEST

TABLE 1-continued

*ADMIN Server
<pre> // CASE statement as long as number of rules in rules list // rule is applied to path part of URL SELECT RULE CASE: PROTECT   DETERMINE PROTECTION SETUP SUB-DIRECTIVES   APPLY SUB-DIRECTIVE POLICIES   CHECK CREDENTIALS AND AUTHENTICATE USER   IF USER IS AUTHENTICATED     CONTINUE PROCESSING THROUGH RULES     LIST   ELSE, SEND BACK AUTHENTICATION FAILURE CASE: PASS   LOCATE RESOURCE SPECIFIED IN THE PATH PART   OF THE URL, GET READY TO SEND RESOURCE CASE: FAIL   GENERATE FORBIDDEN BY RULE MESSAGE CASE: EXEC   LOCATE PROGRAM TO RUN (IN THE URL)   PREP ENVIRONMENT BY SETTING UP STDIN,   ENVIRONMENT VARIABLES AND STDOUT   RUN PROGRAM   WHEN PROGRAM RETURNS, DATA IS IN STDOUT CASE: MAP   FOR MATCHED ITEM IN TEMPLATE FIELD     REPLACE MATCHED ITEM WITH     REPLACEMENT FIELD   CONTINUE PROCESSING THROUGH RULES LIST // end of processing through rules list, resource requested // should be buffered and present, therefore get ready to // send it to the browser // must be HTTP like before returning data to browser // that includes things such as choosing correct MIME types, // counting output bytes, time of day, server response // headers   APPLY THE CORRECT HTTP RESPONSE HEADERS   RETURN OUTPUT TO BROWSER   CLOSE THIS CONNECTION // note, socket options allow the application to close the // connection without worrying about how much data was sent // (since this is TCP)   RETURN THIS THREAD 330 TO WORKER THREAD POOL   WAIT FOR MORE WORK // end of pseudo code for Server 310 </pre>

The services table referred to at line 33 is used by many of the components of the IBM AS/400 TCP/IP, and is not shown in the figures. At lines 66–67 reference is made to variables STDIN and STDOUT. These are pathways used by the server to receive or send information from/to the CGI programs, with STDIN to the CGI and STDOUT from the CGI.

Referring to Table 2, a sample instance file 318 is set forth. This sample instance file 318 indicates what configuration files 314, 316 are read and processed in step 324 by \*ADMIN server 310. There can be other things in this instance file, and such things would be placed here when the "Instance Parameter" form 420 (FIG. 8) has been correctly filled out. The two files indicated here at lines 2 and 3 would be read and parsed by the \*ADMIN server 310 in step 322. The other flag shown at line 3, -AutoStartN, indicates that this sample server 310 will not Autostart when its TCP/IP stack is invoked. (However, admin server 310 can be autostarted by changing its instance parameter values.) The -r show which configuration file members are being read by the web server for a particular instance (the read-only before the read-write). Line 4 shows that the instance is not autostarted.

TABLE 2

Instance File 318
<pre> 5 // Sample of instance file that drives the admin server 310 -r /QSYS.LIB/QTCP.LIB/QATMHTTP.FILE/DEFAULT.MBR -r /QSYS.LIB/QUSR.SYS.LIB/QATMHTTPA.FILE/ADMIN.MBR -AutoStartN </pre>
<p>10 Referring to Table 3, a sample configuration file is presented for controlling the operation of *ADMIN server 310. In this table, a "#" indicates a comment line, and these are not processed by the server. Other lines set forth the directives that get processed into "rules". These rules are then applied on a per request basis to the path portion of the URL. Following is a description of the functions provided by configuration file 314, including the manner in which to interpret the Enable, Protect, Map, Pass, Exec and AddType commands.</p> <p>DirAccess (Table 3, Line 13)</p> <p>This directive specifies whether the server is to return directory listings when requested. The values on the Welcome and AlwaysWelcome directives determine when a request is interpreted as a request for a directory listing. The default value is Off, which means that the server cannot return directory listings for any directories and subdirectories. If it is desired to control which directories and subdirectories the server can return directory listings for, use:</p> <p>DirAccess Selective</p> <p>If the value is changed to On, the server will return directory listings.</p> <p>If the value is changed to Selective, the server will return directory listings for any directory that contains a file named wwwbrws object. The contents of the wwwbrws file are not important, the server only checks for its existence. The object is a member name of an AS/400 physical file or a type of object in an integrated file system directory. For case-sensitive file systems such as /QOpenSys, the wwwbrws name is lower-case.</p> <p>Examples:</p> <p>DirAccess On</p> <p>DirAccess Selective</p> <p>Enable Line 15</p> <p>This directive is used to specify which HTTP methods the server is to accept. As many of the HTTP methods may be enabled as needed. For each method the server is to accept, a separate Enable directive is entered followed by the name of the method.</p> <p>Example:</p> <p>Enable POST</p> <p>Protection Table 3 Lines 21–31</p> <p>This directive is used to define a protection setup within the configuration file. The protection setup is given a name and the type of protection is defined using protection sub-directives.</p> <p>Note:</p> <p>In the configuration file, Protection directives must be placed before any DefProt or Protect directives that point to them.</p>

The format of the directive is:

---

```
Protection label-name {
    subdirective value
    subdirective value
    ...
}
```

---

where:

#### Label-name

The name to be associated with this protection setup. The name can then be used by subsequent DefProt and Protect directives to point to this protection setup.

#### Subdirective Value

A protection subdirective and its value is placed on each line between the left brace and the right brace. No comment lines may appear between the braces.

See "Protection Subdirectives" for descriptions of the protection subdirectives.

Example:

---

```
Protection NAME-ME {
    AuthType Basic
    ServerID restricted
    PasswdFile QUSRSPYS/WWW/PASSWORD
    GroupFile /WWW/group.grp
    GetMask groupname
}
```

---

#### Protection Subdirectives

Following are descriptions of each of the protection subdirectives that can be used in a protection setup. The subdirectives are in alphabetical order.

Protection setups are within the configuration file as part of DefProt, Protect, or Protection directives.

A Mask, GetMask, or PostMask subdirective must be specified in the protection setup.

See "Protection example" and the previous descriptions of the DefProt, Protect, and Protection directives for examples of using protection setups.

#### ACLOverride

Specifies that ACL Files Override Protection Setups.1.

This subdirective is used with a value of On if Access Control List files (ACL) is to override the masks specified in the protection setup. If a directory being protected by the protection setup has an ACL file, the mask subdirectives in the protection setup are ignored. (The mask subdirectives are GetMask, Mask, and PostMask.)

See "Using Access Control List (ACL) files" for more information on ACL files.

Example:

#### ACLOverride On

#### AuthType

Specifies Authentication Type.

This subdirective is used to limit access based on user names and passwords. The type of authentication to use when the client sends a password to the server is specified. With basic authentication (AuthType Basic), passwords are sent to the server as plain text. They are encoded, but not encrypted. Only basic authentication is supported.

Example:

AuthType Basic

#### GetMask

Specifies the User names, Groups, and Addresses Allowed to get Files.

This subdirective is used to specify user names, groups, and address templates authorized to make GET requests to a protected directory. The special value all@(\*) will allow all requesters access. See "Rules for specifying user names, group names, and address templates".

Example:

10      GetMask authors,(niceguy,goodie)@96.96.3.1,
 128.141.\*.\*

#### GroupFile

Specifies the Location of the Associated Group File.

This subdirective is used to specify the path and file name of the server group file that this protection setup is to use.

15      The groups defined within the server group file can then be used by:

1. Any mask subdirectives that are part of the protection setup. (The mask subdirectives are GetMask, Mask, and PostMask.)

20      2. Any ACL file on a directory that is protected by the protection setup.

See "Using server group files" more information about server group files.

Example:

25      GroupFile /docs/etc/WWW/restrict.grp
 Mask

Specifies the User Names, Groups, and Addresses Allowed to Make HTTP Requests.

This subdirective is used to specify user names, groups, and address templates authorized to make HTTP requests not covered by other mask subdirectives. The special value all@(\*) will allow all requesters access. See "Rules for specifying user names, group names, and address templates". See "Methods—Set method acceptance" for descriptions of the HTTP methods supported by the server. Note:

Masks are case sensitive. The following is an example of how Mask protection is issued on a user ID:

Example:

MASK WEBADM,webadm
 PasswdFile

Specifies the Location of the Associated Validation List.

This subdirective is used when limiting access based on user names and passwords. The PasswdFile directive may be issued by one of the following methods:

45      1. Specify the path name of the validation list that this protection setup is to use:
 libname/validation\_list\_name

2. Specify %%SYSTEM%% to indicate that the AS/400 user profiles are to be used to validate user names and passwords.

50      Examples:

PasswdFile %%SYSTEM%%
 PasswdFile QUSRSPYS/HEROES

In the above example, note that %%SYSTEM%% lets the server know that password verification is done with an AS/400 user profile.

#### PostMask

Specifies the User Names, Groups, and Addresses Allowed to Post Files.

60      60      For a secure server, this subdirective is used to specify users, groups, and address templates authorized to make POST requests to a protected directory. The special value all@(\*) will allow all requesters access. See "Rules for specifying user names, group names, and address templates".

Example:

PostMask Anyone@9.136.\*

**ServerID**

Specifies a Name to Associate with the Validation List.

This subdirective is used when limiting access based on user names and passwords. A name to associate with the validation list being used is specified. The name does not need to be a real machine name.

The name is used as an identifier to the requester. Since different protection setups can use different validation lists, having a name associated with the protection setup can help the client decide which password to send. This name may be displayed when prompting for a user name and password.

Example:

**ServerID restricted**

**UserId**

Specifies the Access Control User ID that the Server should Use.

This subdirective is used to specify the AS/400 user profile that the server switches to while completing the HTTP transaction.

**%%SYSTEM%%** is specified to use the profile of the server, default QTMHHTTP.

**%%CLIENT%%** is specified to use the user profile that was supplied when challenged for user ID and password.

The values specified here (on the UserId Protection subdirective) override the values specified on the UserId directive.

Example:

**UserId WWW**

Protect Table 3 Line 37

This directive is used to activate protection setup rules for requests that match a template.

For protection to work properly, DefProt and Protect directives must be placed before any Pass or Exec directives in the configuration file.

The format of the directive is different depending upon whether it is to point to a label containing the protection subdirectives or to include the protection subdirectives as part of the Protect directive.

1. To point to a label containing the protection subdirectives, the format is as follows:

**Protect request-template [label [FOR IP-address-template]]**

2. To include the protection subdirectives as part of the Protect directive, the format is as follows:

---

**Protect request-template [IP-address-template] {**  
        **subdirective value**  
        **subdirective value**

---

}

---

where:

**Request-template**

A template for requests for which protection is to be activated. The server compares incoming client requests to the template and activates protection if there is a match.

**Label**

This parameter is used to identify the protection setup containing the protection subdirectives to activate for requests that match request-template.

This parameter is optional. If it is omitted, the protection setup is defined by the most recent DefProt directive that contains a matching template.

Protection setup is defined with protection subdirectives.

See "Protection Subdirectives" for descriptions of the protection subdirectives.

A protection setup label name that matches a name defined earlier on a Protection directive. The Protection directive contains the protection subdirectives.

**Subdirective Value**

A parameter used to include the protection subdirectives as part of the Protect directive. The left brace character must be the last character on the same line as the Protect directive. Each subdirective follows on its own line. The right brace character must be on its own line following the last subdirective line.

10     No comment lines may be placed between the braces.

See "Protection Subdirectives" for descriptions of the protection subdirectives.

**For IP-address-template**

If the server has multiple connections, this parameter can be used to specify an address template. The server uses the directive only for requests that come to the server on a connection with an address matching the template. It is the address of the server's connection that is compared to the template, not the address of the requesting client.

15     A complete IP address may be specified (for example, 204.146.167.72) Or, an asterisk can be used as a wildcard character to specify a template (for example, 9.99.\*).

This parameter is optional. Without this parameter, the server uses the directive for all requests regardless of the connection the requests come in on.

20     To use this parameter, the label or subdirective value parameters must also be used.

Examples:

---

```
30            UserId ANYBODY
          Protection DEF-PROT {
            UserId       BUSYBODY
            AuthType Basic
            ServerID restricted
            PasswdFile   QUSRYS1/WWW/restrict
            GroupFile    /docs/WWW/restrict.grp
            GetMask authors
          }
          Protect /secret/business/*   DEF-PROT
          Protect /topsecret/*   {
            AuthType Basic
            ServerID restricted
            PasswdFile   QUSRYS/TOPBRASS
            GroupFile    /docs/WWW/restrict.grp
            GetMask topbrass
          }
          Pass    /secret/scoop/*    /WWW/restricted/*
          Pass    /secret/business/*   /WWW/confidential/*
          Pass    /topsecret/*    /WWW/topsecret/*
```

---

50 In the above example, the server would activate protection as follows:

1. Since the Protect directive does not specify a protection setup, the protection setup on the previously matching DefProt directive is used. Also, the server changes to the user profile of webpage as defined on the DefProt directive.

2. Requests beginning with /secret/business/ activate protection. The protection setup is defined on the Protection directive that has a label of DEF-PROT. Also, the server changes to the user profile of busybody as defined in the DEF-PROT protection setup.

3. Requests beginning with /topsecret/ activate protection. The protection setup is included directly on the Protect directive. The user profile defaults to ANYBODY. (ANYBODY comes from the UserId directive at the beginning of the example.)

Note: The user profile ANYBODY must exist and the server much have authority to use it.

Examples:

---

```

Protect  /secret/*  CustomerA-PROT webname
204.146.167.72
Protect  /secret/*  CustomerB-PROT webname 9.67.*
Protect  /topsecret/*  webname 204.146.167.72 {
    AuthType Basic
    ServerID restricted
    PasswdFile  QUSRYS/WWW/CUSTOMERA
    GroupFile   /docs/WWW/customer-A.grp
    GetMask A-brass
}
Protect  /topsecret/*  webname 9.67.*      {
    AuthType Basic
    ServerID restricted
    PasswdFile QUSRYS/WWW/CUSTOMERB
    GroupFile /docs/WWW/customer-B.grp
    GetMask B-brass
}

```

---

The above examples use the optional IP address template parameter. If the server receives requests that begin with /secret/ or /topsecret/, it activates a different protection setup for the request based on the IP address of the connection the request comes in on.

For /secret/ requests coming in on 204.146.167.72, the server activates the protection setup defined on a Protection directive with a label of CustomerA-PROT. For /topsecret/ requests coming in on 204.146.167.72, the server activates the protection setup defined inline on the first Protect directive for /topsecret/.

For /secret/ requests coming in on any connection with an address beginning 9.67, the server activates the protection setup defined on a Protection directive with a label of CustomerB-PROT. For /topsecret/ requests coming in on any connection with an address beginning 9.67, the server activates the protection setup defined inline on the second Protect directive for /topsecret/.

Map Table 3 Line 49

This directive is used to specify a template for requests change to a new request string. After the server changes the request, it takes the new request string and compares it to the request templates on subsequent directives.

The format of the directive is:

Map request-template new-request [IP-address-template]  
where:

#### Request-template

A template for requests that the server is to change and then continue comparing the new request string to other templates.

An asterisk may be used as a wildcard in the template.

#### New-request

The new request string the server is to continue to compare to the request templates on subsequent directives. new-request may contain a wildcard if the request-template has one. The part of the request that matches the request-template wildcard is inserted in place of the wildcard in new-request.

#### IP-address-template

If the server has multiple connections, this parameter can be used to specify an address template. The server uses the directive only for requests that come to the server on a connection with an address matching the template. It is the address of the server's connection that is compared to the template, not the address of the requesting client.

A complete IP address may be specified (for example, 65 204.146.167.72). Or, an asterisk may be used as a wildcard character and a template specified (for example, 9.99.\*).

This parameter is optional. Without this parameter, the server uses the directive for all requests regardless of the connection the requests come in on.

Examples:

Example for a Map request with /cgi-bin/ as a PGM object:

---

```

10  Map /cgi-bin/*  /cgi-bin/*.pgm
    Exec /cgi-bin/*.pgm  /qsys.lib/cgilib.lib/*

```

---

In the above example, the server would take any requests starting with /cgi-bin/ and change the /cgi-bin/ portion of the request to /cgi-bin/\*.pgm. Anything that followed /cgi-bin/ on the original request would also be included in the new request string. So /cgi-bin/whatsup/ would change to /cgi-bin/whatsup.pgm.

The Exec directive would change the /cgi-bin/whatsup.pgm to /qsys.lib/cgilib.lib/\*. This change allows 20 the Internet Connection Secure Server to execute the program cgipgm in library cgidb. The server would take the new request string and continue to compare it to request templates on subsequent directives.

---

```

25  Map  /stuff/*  /customerA/good/stuff/*  204.146.167.72
    Map  /stuff/*  /customerB/good/stuff/*  9.99./*

```

---

30 The above examples use the optional IP address template parameter. If the server receives requests that begin with /stuff/, it changes the request to a different request string based on the IP address of the connection the request comes in on. For requests coming in on 204.146.167.72 the server changes the /stuff/ portion of the request to /customerA/good/stuff/. For requests coming in on any connection with an address beginning 9.99, the server changes the /stuff/ portion of the request to /customerB/good/stuff/.

Pass Table 3 Line 76

This directive is used to specify a template for requests to 40 be accepted and responded to with a document from the server. Once a request matches a template on a Pass directive, the request is not compared to request templates on any subsequent directives.

The format of the directive is:

45 Pass request-template [file-path IP-address-template]]  
where:

#### Request-template

A template for requests the server is to accept and respond to with a document from the server.

50 An asterisk can be used as a wildcard in the template.  
File-path

The path to the file that contains the document the server is to return. file-path may contain a wildcard if the request-template has one. The part of the request that matches the request-template wildcard is inserted in place of the wildcard in file-path.

This parameter is optional. If a path is not specified, the request itself is used as the path.

#### IP-address-template

55 If the server has multiple connections, this parameter can be used to specify an address template. The server uses the directive only for requests that come to the server on a connection with an address matching the template. It is the address of the server's connection that is compared to the template, not the address of the requesting client.

A complete IP address may be specified (for example, 65 204.146.167.72). Or, an asterisk can be used as a wildcard character and a template specified (for example, 9.99.\*).

This parameter is optional. To use this parameter, the file-path parameter must also be used. Without the IP-address-template parameter, the server uses the directive for all requests regardless of the connection the requests come in on.

Examples:

QDLS:	Pass	/doc/*	/QDLS/WEBTEST/*.HTM
Root:	Pass	/root/*	/WEBSAMP/*.html
QOpenSys:	Pass	/x/*	/QOpenSys/WEBSAMP/*.html
QSYS:	Pass	/lib/*	/QSYS.LIB/WEBSAMP.LIB/HTMLDOC.FILE/*.MBR

Specifying Pass with /\* as a template and no replacement string allows the server to serve any AS/400 file that QTMHHTTP user profile has read access to, with a request that specifies the AS/400 real document name; for example, /QSYS.LIB/PERSONAL.LIB/DATA.FILE/PAY.MBR. A Pass directive with /\* as a template can be used to refer to a replacement string that would serve an AS/400 document; for example, Pass /\* /www/webdata/web.html. This would be used after all other Map, Pass, Exec, and Redirect directives to prevent anyone from getting an Error 403 "Forbidden by rule".

In the above example for the QDLS file system, the server would respond to a request starting /doc/ with a document from /QDLS/WEBTEST/. Anything that followed /doc/ would also be used to identify the document. So the server would respond to the request /doc/test/test1/doctest.html with the document in file /QDLS/WEBTEST/test/test1/doctest.html.

Pass /gooddoc/\*

In the above example, the server would respond to a request starting with /gooddoc/ with a document from /gooddoc. So the server would respond to the request /gooddoc/volume1/issue2/newsletter4.html with the document in file /gooddoc/volume1/issue2/newsletter4.html.

Pass	/parts/*	/customerA/catalog/*	204.146.167.72
Pass	/parts/*	/customerB/catalog/*	9.99.*

The above examples use the optional IP address template parameter. If the server receives requests that begin with /parts/, it returns a file from a different directory based on the IP address of the connection on which the request is received. For requests coming in on 204.146.167.72 the server returns a file from /customerA/catalog/. For requests coming in on any connection with an address beginning 9.99, the server returns a file from /customerB/catalog/.

Exec Table 3 Line 126

This directive is used to specify a template for requests to be accepted and responded to by running a CGI program. Once a request matches a template on an Exec directive, the request is not compared to request templates on any subsequent directives.

The format of the directive is:

Exec request-template program-path [IP-address-template] where:

Request-template

A template for requests that the server is to accept and respond to by running a CGI program.

An asterisk must be used as a wildcard in the program-path. The part of the request that matches the request-template wildcard must begin with the name of the file that

contains the CGI program. The request-template is case sensitive, but the replacement string is only case sensitive when it refers to a case sensitive file system.

Example:

5

Map	/cgi-bin/*	/cgi-bin/*.pgm
Exec	/cgi-bin/*	/qsys.lib/cgilib.lib/*

10 This example substitutes the value after /cgi-bin/ as the name of the program. The Exec directive identifies the library where the CGI program specified in /cgi-bin/ is stored. The Exec directive changes the directive to the cgilib library.

15 Program-path

The path to the file that contains the CGI program that the server is to execute for the request.

CGI programs can be run from the QSYS file system, only. program-path must also contain a wildcard. The wildcard is replaced with the name of the file that contains the CGI program. The request can also contain additional data that is passed to the CGI program in the PATH\_INFO environment variable. The additional data follows the first slash character that comes after the CGI program file name on the request. The data is passed according to CGI specifications.

25 IP-address-template

If the server has multiple connections, this parameter is used to specify an address template. The server uses the directive only for requests that come to the server on a connection with an address matching the template. It is the address of the server's connection that is compared to the template, not the address of the requesting client.

A complete IP address may be specified (for example, 204.146.167.72). Or, an asterisk may be used as a wildcard character and a template specified (for example, 9.99.\*).

30 This parameter is optional. Without this parameter, the server uses the directive for all requests regardless of the connection the requests come in on.

Example

40

Map	/cgi-bin/*	/cgi-bin/*.pgm
Exec	/cgi-bin/*	/QSYS.LIB/COIBIN.LIB/*

45 In the above example, the server expects to find the CGI program in library CGIBIN. In this case, the request for http://hostname/cgi-bin/mycgi causes the server to attempt to run the program named MYCGI, in library CGIBIN. The Map directive adds the .pgm to the program name so .pgm does not have to be specified on the URL request. The request arrives at the server as:

55 /cgi-bin/mycgi /cgi-bin/mycgi.pgm /QSYS.LIB/CGIBIN.LIB/*	The request arrives. The Map directive adds the .pgm. The request changes to identify the path to the program (AS/400 library.)
---	---

60 AddType Table 3 Line 132

This directive is used to bind files with a particular extension to a MIME type/subtype. Multiple occurrences of this directive may be used in the configuration file. The format of the directive is:

65 AddType extension type/subtype encoding [quality] where:  
.extension

The file extension pattern. The wildcard character (\*) may be used only on the following two special extension patterns:

\*\*

Matches all file names that contain a dot character (.) and have not been matched by other rules

\*

Matches all file names that do not contain a dot character (.) and have not been matched by other rules

Type/Subtype

The MIME type and subtype to bind to files that match the corresponding extension pattern.

Encoding

The MIME content encoding to which the data has been converted:

7 bit

Data is all represented as short (less than 1000 characters) lines of US-ASCII data. Source code or plain text files usually fall into this category. Exceptions would be files containing line-drawing characters or accented characters.

8 bit

Data is represented as short lines, but may contain characters with the high bit set (for example, line-drawing characters or accented characters). PostScript files and text files from European sites usually fall into this category.

binary

This encoding can be used for all data types. Data may contain not only non-ASCII characters, but also long (greater than 1000 characters) lines. Almost every file of type image/\*, audio/\*, and video/\* falls into this category, as do binary data files of type application/\*.

quality

An optional indicator of relative value (on a scale of 0.0 to 1.0) for the content type. The quality value is used if multiple representations of a file are matched by a request. The server selects the file that is associated with the highest quality value. For example, if the file internet.ps is requested, and the server has the following AddType directives:

AddType	.ps	application/postscript	8bit	1.0
AddType	*.*	application/binary	binary	0.3

the server would use the application/postscript line because its quality number is higher.

Example:

AddType .bin application/octet-stream binary 0.8

TABLE 3

Configuration File 314

4 # start sample of configuration file that drives #ADMIN	#
5 # server 310	#
6 # *****	#
7 #	#
8 # HTTP Administration server configuration	#
9 #	#
10 # *****	#
11 #	#
12 # HostName your.full.host.name	#
13 # The default ports for the administration server are set	#
14 # in the services table.	#
15 #	#
16 DirAccess Off	
17 #	#

TABLE 3-continued

Configuration File 314		
18 # ENABLE start		#
19 Enable	POST	
20 Enable	GET	
21 Enable	HEAD	
22 # ENABLE end		#
23 #		#
24 # PROTECTION start - Force authentication and treat as AS/400 *USRPRF		#
25 Protection IBMDFTP {		
26 AuthType Basic		
27 ServerID OS400ADM		
28 Userid %%CLIENT%%		
29 PasswdFile %%SYSTEM%%		
30 GetMask All		
31 PostMask All		
32 }		
33 # PROTECTION end		#
34 #		#
35 # PROTECT start		#
36 Protect /QIBM/Firewall/*	IBMDFTP	
37 Protect /QIBM/NetC/*	IBMDFTP	
38 Protect /	IBMDFTP	
39 Protect /AS400TASKS	IBMDFTP	
40 Protect /QIBM/OS400/*	IBMDFTP	
41 Protect /QIBM/ICS/HTML/*	IBMDFTP	
42 Protect /QSYS.LIB/QSYSCGLIB/*	IBMDFTP	
43 Protect /QSYS.LIB/QTCCPGLLIB/*	IBMDFTP	
44 Protect /QSYS.LIB/QTCPGGLIB/*	IBMDFTP	
45 Protect /QIBM/ICSS/Cert/*	IBMDFTP	
46 # PROTECT end		#
47 #		#
48 # MAP start		#
49 Map /QIBM/Firewall/Admin/*		
50 /QSYS.LIB/QSYSCGLIB/DB2WWW.PGM/QIBM		
51 /ProdData/HTTP/Protect/Firewall/Macro/*		
52 Map /QIBM/NetC/Admin/QSYS.LIB/QSYSCGLIB		
53 /WEBCONFIG.PGM/ladminl		
54 Map /QIBM/NetC/Admin/webconfig/*		
55 /QSYS.LIB/QSYSCGLIB/WEBCONFIG.PGM/*		
56 Map /QIBM/NetworkStation/Admin /QYTC/QYTCMAIN.PGM		
57 Map /QIBM/ICSS/Cert/Admin/*		
58 /QSYS.LIB/QSYSCGLIB/DB2WWW.PGM		
59 /QIBM/ProdData/HTTP/Protect/ICSS/Cert/Macro/*		
60 Map /		
61 /QSYS.LIB/QSYSCGLIB/DB2WWW.PGM		
62 /QIBM/ProdData/HTTP/Protect		
63 /OS400/Macro/qyunmain.ndm/main0		
64 Map /AS400TASKS /QSYS.LIB/QSYSCGLIB/		
65 DB2WWW.PGM/QIBM/ProdData/HTTP/Protect/OS400/Macro		
66 /qyunmain.ndm/main0		
67 Map /QIBM/ICS/HTML/MRI2924/rzag1mst.html		
68 /QSYS.LIB/QTCCPGLLIB/QTMHFMN.PGM/QUICKBGN		
69 Map /QIBM/ICS/HTML/MRI2924/sample.html		
70 /QSYS.LIB/QTCCPGLLIB/QTMHFMN.PGM/SAMPLE		
71 Map /QIBM/ICS/HTML/MRI2924/icswgms1.html		
72 /QSYS.LIB/QTCCPGLLIB/QTMHFMN.PGM/WEBMSTGD		
73 # MAP end		#
74 #		#
75 # PASS start		#
76 Pass /QIBM/OS400/*		
77 /QIBM/ProdData/HTTP/Protect/OS400/HTML/*		
78 Pass /QIBM/Firewall/*		
79 /QIBM/ProdData/HTTP/Protect/Firewall/HTML/*		
80 Pass /QIBM/NetC/*		
81 /QIBM/ProdData/HTTP/Protect/NetC/*		
82 Pass /QIBM/NetworkStation/*		
83 /QIBM/ProdData/HTTP/Protect/NetworkStation/*		
84 #		
85 # Pass statements for misplaced GIF files		
86 #		
87 Pass /QIBM/ICS/HTML/MRI/*docmast.*		
88 /QIBM/ProdData/HTTP/Protect/TC1/ICSS		
89 /HTML/ICONS/docmast.gif		
90 Pass /QIBM/ICS/HTML/MRI/*ics10s03.*		
91 /QIBM/ProdData/HTTP/Protect/TC1/ICSS		
92 /HTML/ICONS/ics10s03.gif		
93 Pass /QIBM/ICS/HTML/MRI/*ics10s04.*		
94 /QIBM/ProdData/HTTP/Protect/TC1/ICSS		

TABLE 3-continued

Configuration File 314	
95 /HTML/ICONS/ics10s04.gif	5
96 Pass /QIBM/ICS/HTML/MRI*/ics10s05.*	
97 /QIBM/ProdData/HTTP/Protect/TC1/ICSS	
98 /HTML/ICONS/ics10s05.gif	
99 Pass /QIBM/ICS/HTML/MRI*/ics10s06.*	
100 /QIBM/ProdData/HTTP/Protect/TC1/ICSS	
101 /HTML/ICONS/ics10s06.gif	
102 Pass /QIBM/TCS/HTML/MRI*/ics10s07.*	10
103 /QIBM/ProdData/HTTP/Protect/TC1/ICSS	
104 /HTML/ICONS/ics10s07.gif	
105 Pass /QIBM/ICS/HTML/MRI*/ics10s08.*	
106 /QIBM/ProdData/HTTP/Protect/TC1/ICSS	
107 /HTML/ICONS/ics10s08.gif	15
108 Pass /QIBM/ICS/HTML/MRI*/ics10s14.*	
109 /QIBM/ProdData/HTTP/Protect/TC1/ICSS	
110 /HTML/ICONS/ics10s14.gif	
111 Pass /QIBM/ICS/HTML/MRI*/ICFICON.*	
112 /QIBM/ProdData/HTTP/Protect/TC1/ICSS	
113 /HTML/ICONS/icficon.gif	
114 #	20
115 #	
116 Pass /QIBM/ICS/HTML/*	
117 /QIBM/ProdData/HTTP/Protect/TC1/ICSS/HTML/*	
118 Pass /QIBM/ProdData/HTTP/Public/*	
119 Pass /QIBM/ICSS/Cert/Download/*	25
120 /QIBM/UserData/ICSS/Cert/Download/*	
121 Pass /QIBM/ICSS/Cert/*	
122 /QIBM/ProdData/HTTP/Protect/ICSS/Cert/HTML/*	
123 # PASS end	#
124 #	#
125 # EXEC start	#
126 Exec /QSYS.LIB/QSYSCGI.LIB/*	30
127 Exec /QSYS.LIB/QTCPGCI.LIB/*	
128 Exec /QYTC/* /QSYS.LIB/QYTC.LIB/*	
129 # EXEC end	#
130 #	#
131 # AddType start	#
132 AddType .cacrt application/x-x509-ca-cert 7bit 1.0	35
133 AddType .usrcrt application/x-x509-user-cert 7bit 1.0	
134 AddType .CACRT application/x-x509-ca-cert 7bit 1.0	
135 AddType .USRCRT application/x-x509-user-cert 7bit 1.0	
136 # AddType end	#
137 # ----- #	#
138 # end Sample of configuration file that drives the admin	40
139 # server!	

Referring to FIG. 6, the hierarchy of web pages for \*ADMIN server 310 is set forth. Tasks page 334 is the home page for \*ADMIN server 310, and provides links to first page 336 of ICS configuration and administration, first page 338 of DCM configuration and administration, and first page 340 of firewall configuration and administration. Each of pages 336, 338 and 340 provide links to additional pages 342, 344 and 346, respectively.

Referring to FIG. 7, a specific example 350 of task page 334 is shown. Task page 334 includes buttons for accessing additional pages, including button 352 which when activated will bring up an Internet connection server for configuring an HTTP server and SSL; button 354 which when activated will bring up a page for setting up and monitoring an Internet firewall; button 356 which when activated will bring up a page for configuring a net commerce server; and button 358 which when activated will bring up a page for creating, distributing and managing digital certificates.

The purpose of tasks page 350 is to provide a single URL or collection point for all those applications or system components that use the web browser for configuration and administration. As shown here, four applications 352, 354, 356 and 358 appear on tasks page 350.

Referring to Table 4, an HTML representation of a specific example of an AS/400 TASKS page 350 is set forth.

TABLE 4

Tasks Page 150 (HTML)
1 <HTML>
2 <HEAD>
3 <TITLE>AS/400 Tasks</TITLE>
4 <P>
5 <SCRIPT LANGUAGE="JavaScript">
6 <!--
7 function showHelpFunction() {
8 window.open("//QSYS.LIB/QSYSCGI.LIB/
9 DB2WWW.PGM/QIBM/ProdData/HTTP/
10 Protect/OS400/Macro/qyntmain.ndm/help0","","width=600,
11 height=400,menubar=yes,scrollbars=yes,resizable=yes,status")}
12 //-->
13 </SCRIPT>
14 </HEAD>
15 <BODY BGCOLOR="#FFFFFF" >
16 <CENTER>
17 <TABLE WIDTH=80%>
18 <TR>
19 <TD>
20 <FONT SIZE=5>
21 <STRONG>
22 <IMG SRC="/QIBM/OS400/Icons/ibmlogo.gif" ALIGN=LEFT ALT="IBM">
23 </STRONG>
24 </FONT>
25 </td>
26 <TD rowspan=2>
27 <CENTER>
28 <FONT SIZE=6>
29 <STRONG>AS/400 Tasks
30 </STRONG>
31 </FONT>
32 </CENTER>
33 </TD>
34 <TD ALIGN= CENTER>
35 <IMG SRC="/QIBM/OS400/Icons/as400.gif" ALT="">
36 </TD>
37 </TR>
38 <TR>
39 <td>
40 <FONT Size=1>(C) IBM Corporation 1997
41 </FONT>
42 </TD>
43 <td>
44 <center>
45 <FONT Size=2>RS008.ENDERICK.IBM.COM
46 </FONT>
47 </center>
48 </td>
49 </TR>
50 <TR>
51 <TD colspan=3>
52 <HR size=1>
53 </td>
54 </TR>
55 <TR>
56 <td colspan=3>
57 <P>
58  
59 <CENTER>
60 <TABLE>
61 <TR>
62 <TD>
63 <A HREF="/QSYS.LIB/QTCPGCI.LIB/QTMHICFP.PGM">
64 <IMG SRC="/QIBM/ICS/HTML/ICONS/anchgrap.gif" border="0" ALT="">
65 </A>
66 </TD>
67 </TR>
68 <TD>
69 <A HREF="/QSYS.LIB/QTCPGCI.LIB/QTMHICFP.PGM">
70 <STRONG>Internet Connection Server for AS/400
71 </STRONG>
72 </A>
73  
74 <FONT SIZE=2>Configure the AS/400 HTTP Server and SSL
75 </FONT>

TABLE 4-continued

Tasks Page 150 (HTML)	
76 </TD>	149 </font>
77 </TR>	150 </center>
78 <TR>	151 </td>
79 <TD>	152 </TR>
80 <A HREF="/QIBM/Firewall/Admin/qisafw1.ndm/main0">	153 </TABLE>
81 <IMG SRC="/QIBM/Firewall/Icons/qisafw1.gif" border="0" ALT="">	154 </center>
82 </A>	155 </BODY>
83 </TD>	156 </HTML>
84 <TD>	
85 <A HREF="/QIBM/Firewall/Admin/qisafw1.ndm/main0">	
86 <STRONG>IBM Firewall for AS/400	
87 </STRONG>	
88 </A>	
89  	
90 <FONT SIZE=2>Set up and monitor an Internet Firewall	
91 </FONT>	
92 </TD>	
93 </TR>	
94 <TR>	
95 <TD>	
96 <A HREF="/QIBM/NetC/Admin">	
97 <IMG SRC="/QIBM/NetC/Server/Icons/netcomm.gif" border="0" ALT="">	
98 </A>	
99 </TD>	
100 </TD>	
101 <TD>	
102 <A HREF="/QIBM/NetC/Admin">	
103 <STRONG>IBM Net.Commerce for AS/400	
104 </STRONG>	
105 </A>	
106  	
107 <FONT STZE=2>Configure the Net.Commerce Server	
108 </FONT>	
109 </TD>	
110 </TR>	
111 <TR>	
112 <TD>	
113 <A HREF="/QIBM/ICSS/Cert/Admin/qycucm1.ndm/main0">	
114 <IMG SRC="/QIBM/ICSS/Cert/Icons/qycucm1.gif" border="0" ALT="">	
115 </A>	
116 </TD>	
117 <TD>	
118 <A HREF="/QIBM/ICSS/Cert/Admin/qycucm1.ndm/main0">	
119 <STRONG>Digital Certificate Manager	
120 </STRONG>	
121 </A>	
122  	
123 <FONT SIZE=2>Create, distribute, and manage Digital Certificates	
124 </FONT>	
125 </TD>	
126 </TR>	
127 </TABLE>	
128 <CENTER>	
129 <P> 	
130 <td>	
131 </TR>	
132 <TR>	
133 <td>	
134 <a href="http://www.as400.ibm.com/anchor/MRI2924/anchome.htm">	
135 <font size="-1">Related task information	
136 </font>	
137 </a>	
138 </td>	
139 </td>	
140 </td>	
141 </td>	
142 <center>	
143 <a href="Javascript:showHelpFunction()">	
144	
145  Help	
146 </a>	
147  	
148 <font size="-2">(Requires JavaScript)	

TABLE 4-continued

Tasks Page 150 (HTML)	
5	149 </font>
	150 </center>
	151 </td>
	152 </TR>
	153 </TABLE>
	154 </center>
10	155 </BODY>
	156 </HTML>

Referring to FIG. 8, a sample representation of first page 360 of ICS configuration and administration 336 is shown. 15 This is brought up at browser 304 by selecting the Internet Connection Server for AS/400 link 352 on tasks page 350. Internet connection secure server 360 includes buttons 362 for selecting configuration and administration, 364 for selecting a sample home page, 366 for selecting Internet connection family web site, and 368 for selecting As/400 tasks.

Referring to Table 5, an HTML representation for page 360 is set forth. This first page 360 is also referred to as the front page. Tables 5-9 are created by the CGI scripts.

TABLE 5

Front Page (HTML)	
30	1 ## Sample HTML representation of First Page of ICS
	2 ## configuration and Administration
	3 <html>
	4 <head>
	5 <title>Internet Connection Secure Server for AS/400</title>
	6
35	7 </head>
	8 <frameset COLS="25,75">
	9 <frame SRC="/QIBM/ICS/HTML/MRI2924/ICSNAV.HTML"
	10 NAME="ICSNavFrame">
	11 <frameset ROWS="78,22">
	12 <frame SRC="/QIBM/ICS/HTML/MRI2924/ICSSLOGO.HTML"
	13 NAME="ICSSLogoFrame">
40	14 <frame SRC="/QIBM/ICS/HTML/MRI2924/ICSBUTN.HTML"
	15 NAME="ICSBButtonsFrame">
	16 </frameset>
	17 </frameset>
	18 <form NAME="icsframe" ACTION="" METHOD="POST">
	19 <center>
45	20 </form>
	21 </center>
	22 </body>
	23 </html>

50 Referring to FIG. 9, a representation of link configuration and administration page 370 is illustrated. This is a specific example of a page selected by choosing configuration and administration button 362 from page 360 (FIG. 5). This page 370 may be used to add a new ICS server instance, or 55 manage an existing one. To add a new server instance, an instance name is entered in field 386 and add button 388 activated. To manage an existing server instance, the server is chosen by activating ADMIN radio button 372 or DEFAULT radio button 374, and the action then selected 60 and initiated by activating change button 376, delete button 378, start button 380, stop button 382, or restart button 384. The Delete, Start, Stop and Restart functions perform that action and indicate back the results. Selecting change button 376 brings up panel 400 (FIG. 10). The global attribute 65 values may be changed by selecting field 390.

Referring to Table 6, the HTML code used to generate page 370 is set forth.

TABLE 6

## General Configuration and Administration (HTML)

```

1 // Sample HTML representation of "Configuration and
2 // Administration"
3 <html>
4 <head>
5 <title>General Configuration and Administration</title>
6 </head>
7 
9 <strong>Configuration and Administration
10 </strong>
11 <p><br>
12 <body>
13 <br>
14 <h1>General Configuration and Administration</h1>
15 <hr>Choose an existing server instance and an action to
16 perform on it.
17 <br>
18 <form ACTION="/QSYS.LIB/QTCPCGI.LIB/
QTMHFMOU.PGM/instact"
19 METHOD="POST">
20 <input TYPE=RADIO NAME="INSTANCE"
VALUE="ADMIN" ADMIN
21 <BR>
22 <input TYPE=RADIO NAME="INSTANCE"
VALUE="DEFAULT">DEFAULT
23 <BR>
24 <input type="submit" name="pushbutton" value="Change">
25 <input type="submit" name="pushbutton" value="Delete">
26 <input type="submit" name="pushbutton" value="Start">
27 <input type="submit" name="pushbutton" value="Stop">
28 <input type="submit" name="pushbutton" value="Restart">
29 </form>
30 <hr>To generate a new server instance, specify an instance
31 name and select the "Add" button.
32 <br>
33 <form ACTION="/QSYS.LIB/QTCPCGI.LIB/
QTMHFMOU.PGM/instact"
34 METHOD="POST">
35 Instance name <input TYPE="text" NAME="instance" SIZE=10
36 MAXLENGTH=10> <input type="submit" name="pushbutton"
37 value="Add">
38 </form>
39 <hr>Change the
40 <a href="/QSYS.LIB/QTCPCGILIB/QTMHFMIN.PGM/
globatti">
41 Global Attribute Values</a>
42 <pre>
43 <br>
44 <a href="/QSYS.LIB/QTCPCGI.LIB/QTMHICFP.PGM"></a> <a
47 href="/QIBM/ICS/HTML/MRI2924/HLPAS4IM.HTML"></a>
50 </pre>
51 </body>
52 </html>

```

Referring to FIG. 10, a sample representation page 400, the link "CHANGE", displayed in response to activating change button 376 in panel 370, is shown. In this example, on page 370 (FIG. 9), server instance "TEAM42" would have been previously created added through instance name field 386 and button 388. Returning to page 370, it would then show up in the list of instances, and would be selected for change by activating a button similar to default radio button 374 followed by change button 376 to get to page 400.

Server instance page 400 allows specification of the associated configuration file 314 used by this server instance. Radio buttons 406, 408 and 410 are selected to use an existing configuration 402, to create a new configuration or associate a different configuration with this instance, respectively. There are also links 416 and 418 to configu-

ration and administration forms 460 (FIG. 12), and instance parameters forms 420 (FIG. 11), respectively.

Referring to Table 7, the HTML for generating panel 400 responsive to link CHANGE 376 is set forth.

TABLE 7

## Link CHANGE (HTML)

```

5
10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100
105
110
115
120
125
130
135
140
145
150
155
160
165
170
175
180
185
190
195
200
205
210
215
220
225
230
235
240
245
250
255
260
265
270
275
280
285
290
295
300
305
310
315
320
325
330
335
340
345
350
355
360
365
370
375
380
385
390
395
400
405
410
415
420
425
430
435
440
445
450
455
460
465
470
475
480
485
490
495
500
505
510
515
520
525
530
535
540
545
550
555
560
565
570
575
580
585
590
595
600
605
610
615
620
625
630
635
640
645
650
655
660
665
670
675
680
685
690
695
700
705
710
715
720
725
730
735
740
745
750
755
760
765
770
775
780
785
790
795
800
805
810
815
820
825
830
835
840
845
850
855
860
865
870
875
880
885
890
895
900
905
910
915
920
925
930
935
940
945
950
955
960
965
970
975
980
985
990
995
1000
1005
1010
1015
1020
1025
1030
1035
1040
1045
1050
1055
1060
1065
1070
1075
1080
1085
1090
1095
1100
1105
1110
1115
1120
1125
1130
1135
1140
1145
1150
1155
1160
1165
1170
1175
1180
1185
1190
1195
1200
1205
1210
1215
1220
1225
1230
1235
1240
1245
1250
1255
1260
1265
1270
1275
1280
1285
1290
1295
1300
1305
1310
1315
1320
1325
1330
1335
1340
1345
1350
1355
1360
1365
1370
1375
1380
1385
1390
1395
1400
1405
1410
1415
1420
1425
1430
1435
1440
1445
1450
1455
1460
1465
1470
1475
1480
1485
1490
1495
1500
1505
1510
1515
1520
1525
1530
1535
1540
1545
1550
1555
1560
1565
1570
1575
1580
1585
1590
1595
1600
1605
1610
1615
1620
1625
1630
1635
1640
1645
1650
1655
1660
1665
1670
1675
1680
1685
1690
1695
1700
1705
1710
1715
1720
1725
1730
1735
1740
1745
1750
1755
1760
1765
1770
1775
1780
1785
1790
1795
1800
1805
1810
1815
1820
1825
1830
1835
1840
1845
1850
1855
1860
1865
1870
1875
1880
1885
1890
1895
1900
1905
1910
1915
1920
1925
1930
1935
1940
1945
1950
1955
1960
1965
1970
1975
1980
1985
1990
1995
2000
2005
2010
2015
2020
2025
2030
2035
2040
2045
2050
2055
2060
2065
2070
2075
2080
2085
2090
2095
2100
2105
2110
2115
2120
2125
2130
2135
2140
2145
2150
2155
2160
2165
2170
2175
2180
2185
2190
2195
2200
2205
2210
2215
2220
2225
2230
2235
2240
2245
2250
2255
2260
2265
2270
2275
2280
2285
2290
2295
2300
2305
2310
2315
2320
2325
2330
2335
2340
2345
2350
2355
2360
2365
2370
2375
2380
2385
2390
2395
2400
2405
2410
2415
2420
2425
2430
2435
2440
2445
2450
2455
2460
2465
2470
2475
2480
2485
2490
2495
2500
2505
2510
2515
2520
2525
2530
2535
2540
2545
2550
2555
2560
2565
2570
2575
2580
2585
2590
2595
2600
2605
2610
2615
2620
2625
2630
2635
2640
2645
2650
2655
2660
2665
2670
2675
2680
2685
2690
2695
2700
2705
2710
2715
2720
2725
2730
2735
2740
2745
2750
2755
2760
2765
2770
2775
2780
2785
2790
2795
2800
2805
2810
2815
2820
2825
2830
2835
2840
2845
2850
2855
2860
2865
2870
2875
2880
2885
2890
2895
2900
2905
2910
2915
2920
2925
2930
2935
2940
2945
2950
2955
2960
2965
2970
2975
2980
2985
2990
2995
3000
3005
3010
3015
3020
3025
3030
3035
3040
3045
3050
3055
3060
3065
3070
3075
3080
3085
3090
3095
3100
3105
3110
3115
3120
3125
3130
3135
3140
3145
3150
3155
3160
3165
3170
3175
3180
3185
3190
3195
3200
3205
3210
3215
3220
3225
3230
3235
3240
3245
3250
3255
3260
3265
3270
3275
3280
3285
3290
3295
3300
3305
3310
3315
3320
3325
3330
3335
3340
3345
3350
3355
3360
3365
3370
3375
3380
3385
3390
3395
3400
3405
3410
3415
3420
3425
3430
3435
3440
3445
3450
3455
3460
3465
3470
3475
3480
3485
3490
3495
3500
3505
3510
3515
3520
3525
3530
3535
3540
3545
3550
3555
3560
3565
3570
3575
3580
3585
3590
3595
3600
3605
3610
3615
3620
3625
3630
3635
3640
3645
3650
3655
3660
3665
3670
3675
3680
3685
3690
3695
3700
3705
3710
3715
3720
3725
3730
3735
3740
3745
3750
3755
3760
3765
3770
3775
3780
3785
3790
3795
3800
3805
3810
3815
3820
3825
3830
3835
3840
3845
3850
3855
3860
3865
3870
3875
3880
3885
3890
3895
3900
3905
3910
3915
3920
3925
3930
3935
3940
3945
3950
3955
3960
3965
3970
3975
3980
3985
3990
3995
4000
4005
4010
4015
4020
4025
4030
4035
4040
4045
4050
4055
4060
4065
4070
4075
4080
4085
4090
4095
4100
4105
4110
4115
4120
4125
4130
4135
4140
4145
4150
4155
4160
4165
4170
4175
4180
4185
4190
4195
4200
4205
4210
4215
4220
4225
4230
4235
4240
4245
4250
4255
4260
4265
4270
4275
4280
4285
4290
4295
4300
4305
4310
4315
4320
4325
4330
4335
4340
4345
4350
4355
4360
4365
4370
4375
4380
4385
4390
4395
4400
4405
4410
4415
4420
4425
4430
4435
4440
4445
4450
4455
4460
4465
4470
4475
4480
4485
4490
4495
4500
4505
4510
4515
4520
4525
4530
4535
4540
4545
4550
4555
4560
4565
4570
4575
4580
4585
4590
4595
4600
4605
4610
4615
4620
4625
4630
4635
4640
4645
4650
4655
4660
4665
4670
4675
4680
4685
4690
4695
4700
4705
4710
4715
4720
4725
4730
4735
4740
4745
4750
4755
4760
4765
4770
4775
4780
4785
4790
4795
4800
4805
4810
4815
4820
4825
4830
4835
4840
4845
4850
4855
4860
4865
4870
4875
4880
4885
4890
4895
4900
4905
4910
4915
4920
4925
4930
4935
4940
4945
4950
4955
4960
4965
4970
4975
4980
4985
4990
4995
5000
5005
5010
5015
5020
5025
5030
5035
5040
5045
5050
5055
5060
5065
5070
5075
5080
5085
5090
5095
5100
5105
5110
5115
5120
5125
5130
5135
5140
5145
5150
5155
5160
5165
5170
5175
5180
5185
5190
5195
5200
5205
5210
5215
5220
5225
5230
5235
5240
5245
5250
5255
5260
5265
5270
5275
5280
5285
5290
5295
5300
5305
5310
5315
5320
5325
5330
5335
5340
5345
5350
5355
5360
5365
5370
5375
5380
5385
5390
5395
5400
5405
5410
5415
5420
5425
5430
5435
5440
5445
5450
5455
5460
5465
5470
5475
5480
5485
5490
5495
5500
5505
5510
5515
5520
5525
5530
5535
5540
5545
5550
5555
5560
5565
5570
5575
5580
5585
5590
5595
5600
5605
5610
5615
5620
5625
5630
5635
5640
5645
5650
5655
5660
5665
5670
5675
5680
5685
5690
5695
5700
5705
5710
5715
5720
5725
5730
5735
5740
5745
5750
5755
5760
5765
5770
5775
5780
5785
5790
5795
5800
5805
5810
5815
5820
5825
5830
5835
5840
5845
5850
5855
5860
5865
5870
5875
5880
5885
5890
5895
5900
5905
5910
5915
5920
5925
5930
5935
5940
5945
5950
5955
5960
5965
5970
5975
5980
5985
5990
5995
6000
6005
6010
6015
6020
6025
6030
6035
6040
6045
6050
6055
6060
6065
6070
6075
6080
6085
6090
6095
6100
6105
6110
6115
6120
6125
6130
6135
6140
6145
6150
6155
6160
6165
6170
6175
6180
6185
6190
6195
6200
6205
6210
6215
6220
6225
6230
6235
6240
6245
6250
6255
6260
6265
6270
6275
6280
6285
6290
6295
6300
6305
6310
6315
6320
6325
6330
6335
6340
6345
6350
6355
6360
6365
6370
6375
6380
6385
6390
6395
6400
6405
6410
6415
6420
6425
6430
6435
6440
6445
6450
6455
6460
6465
6470
6475
6480
6485
6490
6495
6500
6505
6510
6515
6520
6525
6530
6535
6540
6545
6550
6555
6560
6565
6570
6575
6580
6585
6590
6595
6600
6605
6610
6615
6620
6625
6630
6635
6640
6645
6650
6655
6660
6665
6670
6675
6680
6685
6690
6695
6700
6705
6710
6715
6720
6725
6730
6735
6740
6745
6750
6755
6760
6765
6770
6775
6780
6785
6790
6795
6800
6805
6810
6815
6820
6825
6830
6835
6840
6845
6850
6855
6860
6865
6870
6875
6880
6885
6890
6895
6900
6905
6910
6915
6920
6925
6930
6935
6940
6945
6950
6955
6960
6965
6970
6975
6980
6985
6990
6995
7000
7005
7010
7015
7020
7025
7030
7035
7040
7045
7050
7055
7060
7065
7070
7075
7080
7085
7090
7095
7100
7105
7110
7115
7120
7125
7130
7135
7140
7145
7150
7155
7160
7165
7170
7175
7180
7185
7190
7195
7200
7205
7210
7215
7220
7225
7230
7235
7240
7245
7250
7255
7260
7265
7270
7275
7280
7285
7290
7295
7300
7305
7310
7315
7320
7325
7330
7335
7340
7345
7350
7355
7360
7365
7370
7375
7380
7385
7390
7395
7400
7405
7410
7415
7420
7425
7430
7435
7440
7445
7450
7455
7460
7465
7470
7475
7480
7485
7490
7495
7500
7505
7510
7515
7520
7525
7530
7535
7540
7545
7550
7555
7560
7565
7570
7575
7580
7585
7590
7595
7600
7605
7610
7615
7620
7625
7630
7635
7640
7645
7650
7655
7660
7665
7670
7675
7680
7685
7690
7695
7700
7705
7710
7715
7720
7725
7730
7735
7740
7745
7750
7755
7760
7765
7770
7775
7780
7785
7790
7795
7800
7805
7810
7815
7820
7825
7830
7835
7840
7845
7850
7855
7860
7865
7870
7875
7880
7885
7890
7895
7900
7905
7910
7915
7920
7925
7930
7935
7940
7945
7950
7955
7960
7965
7970
7975
7980
7985
7990
7995
8000
8005
8010
8015
8020
8025
8030
8035
8040
8045
8050
8055
8060
8065
8070
8075
8080
8085
8090
8095
8100
8105
8110
8115
8120
8125
8130
8135
8140
8145
8150
8155
8160
8165
8170
8175
8180
8185
8190
8195
8200
8205
8210
8215
8220
8225
8230
8235
8240
8245
8250
8255
8260
8265
8270
8275
8280
8285
8290
8295
8300
8305
8310
8315
8320
8325
8330
8335
8340
8345
8350
8355
8360
8365
8370
8375
8380
8385
8390
8395
8400
8405
8410
8415
8420
8425
8430
8435
8440
8445
8450
8455
8460
8465
8470
8475
8480
8485
8490
8495
8500
8505
8510
8515
8520
8525
8530
8535
8540
8545
8550
8555
8560
8565
8570
8575
8580
8585
8590
8595
8600
8605
8610
8615
8620
8625
8630
8635
8640
8645
8650
8655
8660
8665
8670
8675
8680
8685
8690
8695
8700
8705
8710
8715
8720
8725
8730
8735
8740
8745
8750
8755
8760
8765
8770
8775
8780
8785
8790
8795
8800
8805
8810
8815
8820
8825
8830
8835
8840
8845
8850
8855
8860
8865
8870
8875
8880
8885
8890
8895
8900
8905
8910
8915
8920
8925
8930
8935
8940
8945
8950
8955
8960
8965
8970
8975
8980
8985
8990
8995
9000
9005
9010
9015
9020
9025
9030
9035
9040
9045
9050
9055
9060
9065
9070
9075
9080
9085
9090
9095
9100
9105
9110
9115
9120
9125
9130
9135
9140
9145
9150
9155
9160
9165
9170
9175
9180
9185
9190
9195
9200
9205
9210
9215
9220
9225
9230
9235
9240
9245
9250
9255
9260
9265
9270
9275
9280
9285
9290
9295
9300
9305
9310
9315
9320
9325
9330
9335
9340
9345
9350
9355
9360
9365
9370
9375
9380
9385
9390
9395
9400
9405
9410
9415
9420
9425
9430
9435
9440
9445
9450
9455
9460
9465
9470
9475
9480
9485
9490
9495
9500
9505
9510
9515
9520
9525
9530
9535
9540
9545
9550
9555
9560
9565
9570
9575
9580
9585
9590
9595
9600
9605
9610
9615
9620
9625
9630
9635
9640
9645
9650
9655
9660
9665
9670
9675
9680
9685
9690
9695
9700
9705
9710
9715
9720
9725
9730
9735
9740
9745
9750
9755
9760
9765
9770
9775
9780
9785
9790
9795
9800
9805
9810
9815
9820
9825
9830
9835
9840
9845
9850
9855
9860
9865
9870
9875
9880
9885
9890
9895
9900
9905
9910
9915
9920
9925
9930
9935
9940
9945
9950
9955
9960
9965
9970
9975
9980
9985
9990
9995
10000
10005
10010
10015
10020
10025
10030
10035
10040
10045
10050
10055
10060
10065
10070
10075
10080
10085
10090
10095
10100
10105
10110
10115
10120
```

page 420 allows specification of particular "instance" values that can override configuration directives for server 422 for those inputs 424-446 shown on the page. In this specific example, such instance values, or parameters, include autostart 424, number of server jobs minimum 426 and maximum 428, coded character set identifier 430, server mapping tables outgoing 432, 434 and incoming 436, 438, access log file name 440, error log file name 442, non-secure port 444 and secure port 446. For example, specifying a non-secure port in field 444 would override any PORT directive in configuration file 316. These attribute values reflect the contents of a member in a file that is similar to file

318, except instead of being tied to server 310, this file is tied to server 311 (it is modified by server 310, used to control the behavior of server 311). The ADMIN instance is unique in that it uses two config file members. It first reads the one from the read-only file 314, then the one from the read-write file 316.

Referring to Table 8, the HTML code for generating a panel for an instance of TEAM42, which will be like panel 420 of FIG. 11 (which is for a server instance of DEFAULT) is set forth. (In order for Table 8 to produce FIG. 11, "TEAM42" in line 12 would have to be changed to "DEFAULT".)

TABLE 8

Instance Parameters (HTML)	
1 //Sample HTML Representation of link "INSTANCE PARAMETERS"	
2 <html>	
3 <head>	
4 <title>Instance Parameters</title>	
5 </head>	
6  <strong>Configuration and Administration</strong>	
8 <p> 	
9 <body>	
10  	
11 <h1>Instance Parameters</h1>	
12 Server instance: <STRONG>TEAM42</STRONG>	
13 <hr>	
14 <form	
15 ACTION="/QSYS.LIB/QTCPCGI.LIB/QTMHFMOU.PGM/instparm=DEFAULT"	
16 METHOD="POST">	
17 Specify parameter values to be used by this server instance.	
18 <p>	
19 <pre>	
20 Autostart	
21 <select NAME="auto"> <option> NO <OPTION> YES <OPTION	
22 SELECTED> *GLOBAL <SELECT>	
23 Number of server jobs:	
24 Minimum	
25 <input TYPE="text"	
26 NAME="min"	
27 VALUE="*CFG"	
28 maxlength=4	
29 SIZE=4>	
30 Maximum	
31 <input TYPE="text"	
32 NAME="max"	
33 VALUE="*CFG"	
34 maxlength=6	
35 SIZE=6>	
36 Coded character set identifier	
37 <input TYPE="text"	
38 NAME="ccsid"	
39 VALUE="*GLOBAL"	
40 maxlength=7	
41 SIZE=7>	
42 Server mapping tables:	
43 Outgoing EBCDIC/ASCII table	
44 <input TYPE="text"	
45 NAME="outtbl"	
46 VALUE="*GLOBAL"	
47 maxlength=10	
48 SIZE=10>	
49 Library	
50 <input TYPE="text"	
51 NAME="outlib"	
52 VALUE=""	
53 maxlength=10	
54 SIZE=10>	
55 Incoming ASCII/EBCDIC table	
56 <input TYPE="text"	
57 NAME="intbl"	
58 VALUE="*GLOBAL"	
59 maxlength=10	
60 SIZE=10>	
61 Library	

TABLE 8-continued

Instance Parameters (HTML)
<pre> 62 &lt;input TYPE="text" 63   NAME="inlib" 64   VALUE="" 65   maxlength=10 66   SIZE=10&gt; 67 ACCESS log file name 68 &lt;input TYPE="text" 69   NAME="acccfil" 70   VALUE="*CFG" 71   maxlength=512 72   SIZE=50&gt; 73 ERROR log file name 74 &lt;input TYPE="text" 75   NAME="errfil" 76   VALUE="*CFG" 77   maxlength=512 78   SIZE=50&gt; 79 Non-secure port 80 &lt;input TYPE="text" 81   NAME="prt" 82   VALUE="*CFG" 83   maxlength=5 SIZE=5&gt; 84 Secure port 85 &lt;input TYPE="text" 86   NAME="secprt" 87   VALUE="*CFG" 88   maxlength=5 89   SIZE=5&gt; 90 &lt;/pre&gt; 91 &lt;pre&gt; 92 &lt;input type="submit" name="pushbutton" value="Apply"&gt; 93 &lt;input type="reset" name="pushbutton" value="Reset"&gt; 94 &lt;hr&gt; 95 &lt;a href="/OSYS.LIB/QTCPCLIB/QTMHICFP.PGM"&gt; 96 &lt;img align=middle src="/QIBM/ICS/HTML/ICONS/go2first.gif" 97 alt="Front Page "&gt; 98 &lt;/a&gt; 99 &lt;a href="/QIBM/ICS/HTML/MRI2924/HLPAS4IN.HTML"&gt; 100 &lt;img align=middle src="/QIBM/ICS/HTML/ICONS/Skyhelp.gif" 101 alt="Help"&gt; 102 &lt;/a&gt; 103 &lt;/pre&gt; 104 &lt;/form&gt; 105 &lt;/body&gt; 106 &lt;/html&gt;</pre>

Referring to FIG. 12, a sample representation of link configuration and administration forms panel 460, such as is displayed at browser 404 in response to selection of field 416 (FIG. 10) is set forth. This server configuration TEAM42 may be customized by selecting and using one or more of forms 461. These forms include the following: basic form 462 (see form 500, FIG. 13 for a specific example) for specifying required settings; user administration forms 463 for managing user ID and passwords, including add user form 464 for adding a new user ID to a validation list and group file, delete user form 465 for deleting a user from a validation file, check user form 466 for determining if a user exists in a validation list, and change password form 467 for changing a user password in a validation list; directories and welcome page 468 for setting viewing options, including initial page 469 for specifying the welcome pages and directory lists, directory list contents 470 for specifying list columns, and readme text 471 for providing informative directory list text; logging form 472 for customizing the access log and error log, including global log file configuration settings form 473 for specifying log formats, access log file configuration form 474 for specifying access log location, and error log file configuration form 475 for specifying error log location; access control form 476 for setting up access control for the server, including document

protection form 477 for specifying file directories to protect, protection setups form 478 for specifying file protection settings, and access control lists 479 for defining user access files; security form 480 for setting up security for the server, including security configuration form 481 for defining basic parameters for security, create keys form 482 for creating keys and request certificates, receive certificate form 483 for receiving a certificate into the key ring, and key management form 484 for working with keys, root keys, and certificates; resource mapping form 485 for redirecting URLs and defining file extensions, including request routing form 486 for routing URL requests to server files, MIME encodings form 487 for defining encodings and extensions, MIME types form 488 for defining file types and extensions, and languages form 489 for associating language encodings and extensions; timeouts form 490 for closing connections automatically; methods form 491 for setting method acceptance; accessory scripts form 492 for specifying custom method scripts; and performance settings form 493 for defining performance settings, including jobs form 494 for configuring jobs.

Referring to Table 9, HTML coding for setting up configuration and administration forms panel 460 is set forth.

TABLE 9

## Configuration and Administration (HTML)

```

1 // Sample representation of link "Configuration and
2 // Administration Forms"
3 <html>
4 <head>
5 <title>Configuration and Administration Forms</title>
6 </head>
7 <body>
8 
10 <strong>Configuration and Administration</strong>
11 <p><br>
12 <h1>Configuration and Administration Forms</h1>
13 Configuration: <strong>TEAM42</strong>
14 <p>
15 <br>
16 You can customize this server configuration by modifying the
17 forms below:
18 <p>
19 <ul>
20 <li>
21 <a href="/QSYS.LIB/QTCPGILIB/QTMHCFIN.PGM/basic=CONFIG">
22 Basic
23 </a> - Specify required settings
24 <li>
25 <a href="/QSYS.LIB/QTCPGILIB/QTMHCFIN.PGM/useradm=CONFIG">
26 User Administration </a>
27     - Manage user ID and passwords
28 <ul>
29 <li>
30 <a href="/QSYS.LIB/QTCPGILIB/QTMHCFIN.PGM/adduser=CONFIG">
31     Add User </a>
32         - Add a new user ID to a validation list and group file
33 <li>
34 <a href="/QSYS.LIB/QTCPGILIB/QTMHCFIN.PGM/deluser=CONFIG">
35     Delete User </a>
36         - Delete a user from a validation list
37 <li>
38 <a href="/QSYS.LIB/QTCPGILIB/QTMHCFIN.PGM/chkuser=CONFIG">
39     Check User </a>
40         - Determine if a user exists in a validation list
41 <li>
42 <a href="/QSYS.LIB/QTCPGILIB/QTMHCFIN.PGM/chpasswd=CONFIG">
43 Change Password</a>
44         - Change a user password in a validation list
45 <ul>
46 <li>
47 <a href="/QSYS.LIB/QTCPGILIB/QTMHCFIN.PGM/dirapp=CONFIG">
48 Directories and Welcome Page</a>
49         - Set viewing options
50 <ul>
51 <li>
52 <a href="/QSYS.LIB/QTCPGILIB/QTMHCFIN.PGM/initpage=CONFIG">
53 Initial Page </a>
54         - Specify welcome pages and directory lists
55 <li>
56 <a href="/QSYS.LIB/QTCPGILIB/QTMHCFIN.PGM/listcnts=CONFIG">
57 Directory List Contents</a>
58         - Specify list columns
59 <li>
60 <a href="/QSYS.LIB/QTCPGILIB/QTMHCFIN.PGM/dirreadm=CONFIG">
61 Readme Text</a>
62         - Provide informative directory list text
63 </ul>
64 <li><a href="/QSYS.LIB/QTCPGTLIB/QTMHLRIN.PGM/logging=CONFIG">
65 G</a>
66 Logging</a>
67         - Customize access log and error log
68 <ul>
69 <li>
70 <a href="/QSYS.LIB/QTCPGILIB/QTMHLRIN.PGM/logfiles=CONFIG">
71 Global Log File Configuration Settings</a>
72         - Specify log formats
73 <li>
74 <a href="/QSYS.LIB/QTCPGILIB/QTMHLRIN.PGM/accconf=CONFIG">
75     Access Log File Configuration</a>
76         - Specify access log location
77 <li>
```

TABLE 9-continued

Configuration and Administration (HTML)	
<pre> 78 &lt;a href="/QSYS.LIB/QTCPCGLLIB/QTMHLRIN.PGM/erconf=CONFIG"&gt; 79     Error Log File Configuration&lt;/a&gt; 80     - Specify error log location 81 &lt;/ul&gt; 82 &lt;li&gt; 83 &lt;a href="/QSYS.LIB/QTCPCGLLIB/QTMHCFIN.PGM/access=CONFIG"&gt; 84     Access Control&lt;/a&gt; 85     - Set up access control for the server: 86 &lt;/ul&gt; 87 &lt;li&gt; 88 &lt;a href="/QSYS.LIB/QTCPCGLLIB/QTMHCFIN.PGM/docautos=CONFIG"&gt; 89 Document Protection&lt;/a&gt; 90     -Specifyfiledirectorystoprotect 91 &lt;li&gt; 92 &lt;a href="/QSYS.LIB/QTCPCGLLIB/QTMHCFIN.PGM/prtclist=CONFIG"&gt; 93 Protection Setups&lt;/a&gt; 94     - Specify file protection settings 95 &lt;li&gt; 96 &lt;a href="/QSYS.LIB/QTCPCGLLIB/QTMHCFIN.PGM/acldir=CONFIG"&gt; 97     Access Control Lists&lt;/a&gt; 98     - Define user access files 99 &lt;/ul&gt; 100 &lt;li&gt; 101 &lt;a href="/QSYS.LIB/QTCPCGLLIB/QTMHSCIN.PGM/security=CONFIG"&gt; 102 Security&lt;/a&gt; 103     - Set up security for the server: 104 &lt;ul&gt; 105 &lt;li&gt; 106 &lt;a href="/QSYS.LIB/QTCPCGLLIB/QTMHSCIN.PGM/secconf=CONFIG"&gt; 107     Security Configuration&lt;/a&gt; 108     - Define basic parameters for security 109 &lt;li&gt; 110 &lt;a href="/QSYS.LIB/QTCPCGLLIB/QTMHSCIN.PGM/key=CONFIG"&gt; 111     Create Keys&lt;/a&gt; 112     - Create keys and request certificates 113 &lt;li&gt; 114 &lt;a href="/QSYS.LIB/QTCPCGLLIB/QTMHSCIN.PGM/recert=CONFIG"&gt; 115     Receive Certificate&lt;/a&gt; 116     - Receiv a certificate into the key ring 117 &lt;li&gt; 118 &lt;a href="/QSYS.LIB/QTCPCGLLIB/QTMHSCIN.PGM/pwprompt=CONFIG"&gt; 119 Key Management&lt;/a&gt; 120     - Work with keys, root keys, and certificates 121 &lt;/ul&gt; 122 &lt;li&gt; 123 &lt;a href="/QSYS.LIB/QTCPCGLLIB/QTMHCFIN.PGM/resmap=CONFIG"&gt; 124     Resource Mapping&lt;/a&gt; 125     - Redirect URLs and define file extensions 126 &lt;ul&gt; 127 &lt;li&gt; 128 &lt;a href="/QSYS.LIB/QTCPCGLLIB/QTMHCFIN.PGM/mpfrule=CONFIG"&gt; 129     Request Routing&lt;/a&gt; 130     - Route URL requests to server files 131 &lt;li&gt; 132 &lt;a href="/QSYS.LIB/QTCPCGLLIB/QTMHCFIN.PGM/addencod=CONFIG"&gt; 133 MIME Encodings&lt;/a&gt; 134     - Define encodings and extensions 135 &lt;li&gt; 136 &lt;a href="/QSYS.LIB/QTCPCGLLIB/QTMHCFIN.PGM/addtype=CONFIG"&gt; 137     MIME Types&lt;/a&gt; 138     - Define file types and extensions 139 &lt;li&gt; 140 &lt;a href="/QSYS.LIB/QTCPCGLLIB/QTMHCFIN.PGM/addlang=CONFIG"&gt; 141     Languages&lt;/a&gt; 142     - Associate language encodings and extensions 143 &lt;/ul&gt; 144 &lt;li&gt; 145 &lt;a href="/QSYS.LIB/QTCPCGLLIB/QTMHCFIN.PGM/timout=CONFIG"&gt; 146 TimeOuts&lt;/a&gt; 147     - Close connections automatically 148 &lt;li&gt; 149 &lt;a href="/QSYS.LIB/QTCPCGLLIB/QTMHCFIN.PGM/methenab=CONFIG"&gt; 150 Methods&lt;/a&gt; 151     - Set method acceptance 152 &lt;li&gt; 153 &lt;a href="/QSYS.LIB/QTCPCGLLIB/QTMHCFIN.PGM/accscr=CONFIG"&gt; 154 Accessory Scripts&lt;/a&gt;</pre>	

TABLE 9-continued

Configuration and Administration (HTML)	
<pre> 155      - Specify custom method scripts 156 &lt;li&gt;Performance Settings&lt;/a&gt; - Define performance settings: 157 &lt;ul&gt; 158 &lt;li&gt; 159 &lt;a href="/QSYS.LIB/QTCPGQLIB/QTMHCFIN.PGM/performa=CONFIG"&gt; 160 Jobs&lt;/a&gt; 161      - Configure jobs 162 &lt;/ul&gt; 163 &lt;/ul&gt; 164 &lt;hr&gt; 165 &lt;a href="/QSYS.LIB/QTCPGQLIB/QTMHICFP.PGM"&gt; 166 &lt;img src="/QIBM/ICS/HTML/ICONS/go2first.gif" 167 alt="Front Page   "&gt;&lt;/a&gt; 168 &lt;img src="/QIBM/ICS/HTML/ICONS/greycfg.gif" 169 alt="Configuration and Administration Forms   "&gt; 170 &lt;a href="/QIBM/ICS/HTML/MRI2924/sample.html"&gt; 171 &lt;img src="/QIBM/ICS/HTML/ICONS/skyhome.gif" 172 alt="Sample Home Page   "&gt;&lt;/a&gt; 173 &lt;a href="http://www.ics.raleigh.ibm.com/"&gt; 174 &lt;img src="/QIBM/ICS/HTML/ICONS/skyres.gif" 175 alt="Resource List   "&gt;&lt;/a&gt; 176 &lt;a href="/QIBM/ICS/HTML/MRI2924/rzag1inst.html"&gt; 177 &lt;img src="/QIBM/ICS/HTML/ICONS/skydoc.gif" 178 alt="Documentation"&gt;&lt;/a&gt; 179 &lt;/body&gt; 180 &lt;/html&gt; </pre>	

Referring further to FIGS. 1 and 2, if the auto start attribute for administration server 310 is set to \*NO, this server 310 starts only upon command, such as the AS/400 STRTCPSVR(\*HTTP)(\*ADMIN) command. The system administrator may, however, using the browser, configure the default autostart attribute for any server instance 310, 311 to yes, and thus may configure ADMIN server instance 310 so that it will autostart.

Referring again to FIG. 3, to add a new instance of an HTTP server, the user types in an instance name 371 and selects ADD 388. The user will then be shown a confirmation page, can then return to page 400, and select 418 to get the instance parameters page.

Referring further to FIG. 9, the link "Global attribute values" 390 shows an additional page similar to that in FIG. 11 to allow modification of the default HTTP server 311 attributes 323. These default attributes are those which are modified with the CHGHTTPA command, and are shown in FIG. 15, global attribute file 532.

Referring further to FIGS. 10 and 11, a configuration for a particular instance of an HTTP server is illustrated. In such a case, the configuration name is mandatory as this is the configuration used by this instance of the HTTP server. Different instances may use the same or unique configurations. If instances share configurations, and need to co-exist, certain override parameters must be specified. For example, if a port directive is in the configuration, and two instances share that configuration, only one of the instances will start up the other instance will fail unless there is an override parameter in the appropriate PORT field.

Once "apply" 448 is chosen, messages indicate the outcome, which may be either the instance is not changed successfully or the instance is changed. "Reset" 450 resets values back to previously defined values, which may include blank fields.

Referring to FIG. 9, if the link "Global Attribute Values" 390 is selected in panel 370, a page is produced which allows setting of the values of 533, 534, 535, 536, 537, and 538 in FIG. 15.

Referring further to FIG. 10, the sample page illustrated is an indication of what is required on a configuration page.

A system administrator can change attributes for this instance via link 418; designate what configuration file is used via items 402, 404, 406, 408, 410, 412, and 414; and specify configuration file values via link 416.

Referring to FIG. 13, a sample page illustrates basic panel 500. This panel 500 is displayed responsive to selection of link 462 in panel 460, FIG. 12. Instructions 502 may include 35 a statement displayed on the browser in panel 500, such as the following:

"Specify the host name of the computer on which the server is installed, the port number on which the server listens for requests, and the directory the server uses as the root of the data hierarchy (server root). Host name should be a fully qualified name. Default port number should be the well-known HTTP port number (80), or a port number above 1024. Port numbers less than 1024 may be reserved by other programs."

45 Representative data entry fields on page 500 include host name 504, default port number 506, server root 508. Button 510 is activated to look up the host name of requesting clients, button 512 to apply, and 514 to reset.

This panel 500 is representative of the many similar 50 panels which would be displayed by selection in panel 460 of links 463-494.

Referring to FIG. 14, a sample page illustrates the contents of confirmation panel, which is displayed back to the user at browser 304 in response to data entered via basic page 500.

Referring to FIG. 15, secure server instance management files and formats are illustrated for enabling management of multiple server instances.

Referring to FIG. 15 in connection with FIG. 2, in accordance with the preferred embodiment of the invention, one of the server instances is default server instance 311, and another is the ADMIN server 310 instance that is used for the browser interface to the configuration and administration utilities.

60 65 Multiple HTTP server instance management is handled by using instance file (\*PF) 516, configuration file (\*SRCPF) 526, and attribute file (\*PF) 532.

Each member 515, 517, 519, 521 of instance file 516 represents a single HTTP server instance. Instance name 525 is not actually contained within each member of 516, but rather is used to determine which member within file 516 is accessed. Instance file 516 is the same as instance file 318 (FIG. 1), and file 515 is member within file 516. A particular member 515 defines an instance. Each instance (member) 515, 517, 519, 521 contains optional attribute overrides 518, configuration file member name 522, and optional configuration directive overrides 524. The instance (member) name 525 is used to override the values in attribute file 532 for this server instance 515. Configuration file member name 522 identifies, as is represented by line 523, the configuration file member 526 from which server instance 515 reads configuration directives 530 at server startup. Optional configuration directives overrides 524 are used to override configuration directives such as port, sslport, AccessLog and ErrorLog. Each member in configuration file 526 (only one member is shown) represents a set of directives 528, 530 that can be used by server instance 515, 517, 519, 521, at startup time. In accordance with a preferred embodiment of the invention, attribute file 532 includes only one member with fields auto start 533, minjob 534, maxjob 535, ccsid 536, out ccsid 537 and in ccsid 538. Minjob 534 and maxjob 535 attributes map to MinActiveThreads and MaxActiveThreads to enable a job structure to be implemented on a thread model, as is further described in U.S. Pat. No. 6,272,518, filed 17 Aug., 1998(supra).

Referring to FIGS. 16 and 17, the STRTCPSVR 540 and ENDTCPYSR 550 commands are adapted to support multiple server instances. Use of SYRTCPYSR command 540 is an alternative to use of the browser based ADMIN server to create startup values in the instance file.

Referring to FIG. 16, the format of start TCP server command 540 is illustrated, which includes STRTCPSVR field 542, SERVER field 544, RESTART field 546 and HTTPSVR( ) field 548. The start TCP/IP server command 540 is used to start the TCP/IP application servers. The syntax of STRTCPSVR statement 540 is as follows:

STRTCPSVR field 542:	STRTCPSVR
SERVER( ) field 544:	SERVER(   *ALL     *SNMP   *ROUTD   *TELNET   *FTP   *SMTP   *LPD   *WSG   *POP   *HTTP   . . .  )
RESTART field 546:	RESTART(   *NONE   *HTTP   )
HTTPSVR( ) field 548:	HTTPSVR(   *ALL     server-instance-name   *ADMIN     *NONE     instance-startup-values   )

change LPD attributes (CHGLPDA) command. More than one LPD servers may be running.  
 \*WSG specifies the 5250/Hypertext markup language (HTML) workstation gateway (WSG) server is started.  
 \*POP specifies the post office protocol (POP) version 3 servers are started based on the number of servers configured with the change POP attributes (CHGPOPA) command.  
 \*HTTP specifies the world wide web hypertext transfer protocol (HTTP) servers are started based on the number of servers specified in the change HTTP attributes (CHGHTTPA) command.  
 RESTART specifies whether to restart the selected server when the STRTCPSVR command 540 is run. The SERVER parameter 544 must be \*ALL or \*HTTP or this parameter is ignored. \*NONE specifies that no server is to be restarted. \*HTTP specifies that the HTTP server is to be restarted if already running, thus forcing the HTTP server to read in the HTTP configuration and use any configuration values or attributes that have changed since it was last started.  
 HTTPSVR( ) specifies the HTTP server instance to be started as well as any additional startup values to be used by the HTTP server to control the server instance.  
 \*ALL specifies that all sever instances for the HTTP server will be started.  
 server-instance-name specifies the server instance will be started.  
 \*ADMIN specifies that the administration HTTP server will be started. This administration server is an instance of the HTTP server that allows administration of system functions using a web browser.  
 instance-startup-values specifies additional startup values to be used for this server instance. The user is required to have \*IOSYSCFG special authority to specify overrides. These values are used to override previously defined server startup values for the specified server instance, and are as follows:  
 -netcp[nnn] overrides the DefaultNetCp directive.  
 -fscp[nnn] overrides default DefaultFsCP directive.

where:  
 SERVER specifies the TCP/IP application servers to be started by this command.  
 \*ALL specifies that all of the TCP/IP application servers and all HTTP instances are started.  
 \*SNMP specifies that simple network management protocol (SNMP) agent jobs are started.  
 \*ROUTED specifies that the RouteD server is started.  
 \*TELNET specifies that the TELNET server is started. More than one TELNET server job may be running.  
 \*FTP specifies that file transfer protocol (FTP) servers are started based on the number of servers configured with a change FTP attributes (CHGFTPA) command. More than one FTP server job may be running.  
 \*SMTP specifies that the simple mail transfer protocol (SMTP) client and server jobs are started.  
 \*LPD specifies that line printer daemon (LPD) servers are started based on the number of servers configured with the

50 -p[nnn] overrides port directive.  
 -sslpor[nnnn] overrides SSLPort directive.  
 -i[configuration file] overrides configuration file for this instance of the server.  
 -l[log-file-name] same as "AccessLog log-file-name"  
 -newlog[log-file-name] same as "AccessLog log-file-name and Logformat Common".  
 -ddslog[log-file-name] same as "AccessLog log-file-name and Logformat DDS".  
 -erlog[log-file-name] same as "ErrorLog log-file-name".  
 -minat[nn] overrides the MinActiveThreads directive.  
 -maxat[nn] overrides the MaxActiveThreads directive.  
 instance-startup-values specified on the STRTCPSVR \*HTTP command take precedence or can augment configuration data in instance files. Instance files take precedence over or can augment data in configuration files.  
 65 Examples:  
 STRTCPSVR SERVER(\*ALL): starts all of the TCP/IP application servers that have been configured. If the change

FTP attributes (CHGFTPA) command was previously used to configure two FTP servers, both servers are started when STRTCPSVR is issued. Where appropriate, the number of servers to start is based on the number of servers configured for the server being started. The configuration option to automatically start the servers (AUTOSTART) is ignored by the STRTCPSVR command 540. The AUTOSTART parameter is only used by the STRTCP command.

STRTCPSVR SERVER(\*TELNET): starts the TCP/IP TELNET application server. If the TELNET server was previously started, one additional TELNET server job is started.

STRTCPSVR SERVER(\*HTTP) RESTART(\*HTTP): restarts the TCP/IP HTTP application server for all instances of the HTTP server. If the HTTP server was not currently running, then all defined instances of the HTTP server are started.

STARTCPSVR SERVER(\*HTTP) HTTPSVR(htpl): starts the TCP/IP HTTP application server instance named 'htpl' using the startup values previously defined for this server instance.

STRTCPSVR SERVER(\*HTTP) HTTPSVR(HTTP1 '-p 81-sslport 443'): starts the TCP/IP HTTP application server instance named 'htpl', and specifies that the server instance should listen on port 81 for unsecure requests and on port 443 for secure requests. The ports defined here override any previously defined ports used by this server instance.

Referring to FIG. 17, the format of end TCP server command 550 is illustrated, which includes ENDTCP command field 552, SERVERO field 554 and HTTPSVR( ) field 556. ENDTCP command 550 is used to end the TCP/IP application server jobs that are specified in the SERVER parameter 554. If the jobs have any current active connections, these connections are ended immediately. The syntax of the ENDTCP command 550 is as follows:

---

ENDTCP field 552:	ENDTCP
SERVER( ) field 554:	SERVER(  *SNMP   *ROUTED   *TELNET   *FTP   *SMTP   *LPD   *HTTP   *WSG   *POP   ...   *ALL
HTTPSVR( ) field 556:	HTTPSVR(  *ALL   server-instance-name   *ADMIN  )

---

where:

server-instance-name specifies the HTTP server instance to be ended.

Examples:

ENDTCP SERVER(\*ALL) ends all active TCP/IP application server jobs.

ENDTCP SERVER(\*HTTP) HTTPSVR(htpl) ends the TCP/IP HTTP application server instance named 'htpl'.

In a typical startup scenario:

(1) a user initializes operation by installing several files, including default configurations for both the ADMIN HTTP server 310 and default servers 311, and issues a start TCP (STRTCP) which causes the TCP/IP stack to start, and any TCP application that has AUTOSTART=\*YES starts.

(2) However, because the default attribute file 323 has AUTOSTART=\*NO for HTTP servers, and there is, in the preferred embodiment, no override for the ADMIN instance (attribute file 318), the user must run an explicit command to start the ADMIN server 310. (That command is STRTCPSVR 540, infra.) This is true only initially, before the ADMIN server 310 is started for the first time. The user can change the autostart attribute in the ADMIN

server's instance file via browser-based config. If the admin server is not running, the only way to start is with STRTCPSVR, because it is the admin server that provides the browser-based config function. Other instances can then be started thru the browser (which actually does STRTCPSVR under the covers).

(3) In response to the STRTCP command, because server 310 has basic authentication enabled, the user is required to enter a valid user ID and password in order to further configure the server.

(4) Once authorized, the user issues STRTCPSVR 540 or STRTCPSVR(\*HTTP), or points the browser to the ADMIN server and selects "default server" and "start". An HTTP server with a standard default configuration that serves only a "welcome.html" document is started on the standard HTTP port 80 for this host.

(5) The user then configures the server (either by the browser and selecting "which server" and "configure" or WRKHTTPCFG(\*configuration)). Using either method, the user is able to (by configuration name): select a configuration file, and then change that particular configuration file.

Referring to FIG. 18, in accordance with the Internet connection secure server embodiment of the invention, a server can listen on both a secure port 204 using secure sockets layer (SSL) and a non-secure port 202 using sockets for requests coming from HTTP clients on network 200. When a listening thread 206 detects a request coming into port 202 or 204 from network 200, it passes the socket descriptor for the request off to a worker thread 212 or 214 that is waiting in a thread pool 210 of initialized threads. Worker thread 212 or 214 processes the request and completes the transaction by sending a response back to the HTTP client on network 200. Service threads 220 run in the background to handle server utility functions including alarm 222 and log writer 224.

Referring to FIG. 19 in connection with FIG. 18, as part of initialization, main thread 240 processes configuration file 244 directives. The MinActiveThreads and MaxActiveThreads directives are used to set the boundaries of the pool 206, 210, 220 of available threads that can be used by the server 190. Main program thread 240 initializes thread pool 210 and then uses a thread 202 from pool 210 to bind 260 and listens 262 on an IP address port 202. When server 190 is started with directives normalmode on and sslmode on there will be one thread 206, 262 listening on port 80 (the HTTP default port 202) and one thread 206, 272 listening on port 443 (the HTTPS default port 204). Hereafter, these threads may be referred to as listing threads, or parent threads. HTTP and HTTPS default ports 202, 204 can be overridden with a directives port (for (HTTP) and an sslport (for HTTPS). These threads 206 will listen on all IP addresses defined for this host unless the BindSpecific directive is in configuration file 244 and it contains a valid IP address for this host. Then threads 206 only listen on the ports for that IP address.

When a request comes in on either or both ports 202, 204, the listening thread 262, 272 does an accept 264, 274 and passes the incoming request off to a work thread 266, 276 that comes from thread pool 210. Listening thread 262, 272 then goes back to accept() mode 264, 274. The work thread 266, 276 processes the incoming request and sends a response to the client. When work thread 266, 276 completes its tasks it returns to pool 210 of available threads.

Two service threads 220, including log writer thread 224 and alarm thread 222, are spun off at server 190 initialization and wait in the background until their services are needed.

Log writer routine 224 is spun off in its own thread when server 190 starts. It spends most of its time sleeping, but wakes up periodically to write out to log files 226 anything that needs to be logged. It handles writing to the access and error logs.

At initialization, startup job 240, 250 opens the log files 226 that are specified in configuration directives 244 (or overrides) and then passes the file descriptors to the log writer thread 224. As is represented by lines 234 and 236, server listening threads 206 and worker threads 210 add log data to log queue 228 for log writer thread 224 to log. When log writer thread 224 wakes up it checks queue 228 for data to be logged. When there is data to be logged, log writer thread 224 locks the queue with a mutually exclusive (mutex) lock on line 218 and, as is represented by line 238, copies the log data from queue 228 into list 216, zeros out queue 228, and then releases mutex 218. Log writer thread 224 can then log the data from list 216 without having other threads 206, 210 blocked from writing to queue 228.

*ALARM THREAD  
= TIMEOUT  
THREAD IN  
FUNCTION.  
41-19-28*

Alarm thread 222 is spun off (spawned) in its own thread when server 190 starts. It spends most of its time sleeping. It wakes up periodically and checks a list of timers 232 that are associated with active work threads 210, such as 266, 268, 276, 278. When a timer 232 expires it indicates that a work thread 210 had a problem completing its task on time. When alarm thread 222 finds an expired timer the timer is removed from the list of timers 232, the outstanding function being processed is terminated, and the socket 202, 204 being used by the request is closed.

Referring to FIG. 20, in accordance with a specific embodiment of the invention, various AS/400 jobs are started to support an HTTP server 190. A main job 160 is created that will then create additional jobs SSL 180, alarm 182, log 184, and worker 178. Upon properly configuring SSL, a spawn() 162 is issued to create job SSL 180 that will then listen 272 on port 204 for https requests. This job SSL 180 becomes a listening, or parent job to, for example, worker jobs 186. In addition, a number of worker jobs 186 are created (spawn 168) that are kept in a pool and at a later time are dispensed as appropriate with units of work (either requests for documents or to execute CGI programs). Finally, spawn log 166 creates log job 184 and spawn alarm 164 creates alarm job 182. USRSPC 150 is used to share global data across these jobs. USRSPC 150 is a system domain object that is uniquely named per server or worker job instance with the instance name and is located in the QHTTP library.

#### Advantages over the Prior Art

The advantages of the method of the preferred embodiment of this invention include the provision of an improved administration server which serves administration and configuration applications to a browser's graphical user interface.

It is a further advantage of the invention that there is provided an administration server which serves to a browser capability for administering and configuring web enabled system components.

It is a further advantage of the invention that there is provided an improved administration server which is isolated from other servers at the site.

It is a further advantage of the invention that there is provided an improved administration server which is isolated from other servers at the site such that loading of either does not adversely impact the other.

It is a further advantage of the invention that there is provided an improved administration server which enables the management of multiple copies or instances of servers.

It is a further advantage of the invention that there is provided an administration server implemented as a browser.

#### Alternative Embodiments

It will be appreciated that, although specific embodiments of the invention have been described herein for purposes of illustration, various modifications may be made without departing from the spirit and scope of the invention. In particular, it is within the scope of the invention to provide a memory device, such as a transmission medium, magnetic or optical tape or disc, or the like, for storing signals for controlling the operation of a computer according to the method of the invention and/or to structure its components in accordance with the system of the invention.

Accordingly, the scope of protection of this invention is limited only by the following claims and their equivalents.

We claim:

1. Method for serving administration and configuration pages to a web browser, comprising the steps of:  
providing a plurality of instances of servers on a single host system; and

one of said servers being an administration server including a hardened configuration file which is not alterable by way of configuration or administration forms by storing administration and configuration directives in a plurality of files including a read-write configuration file member and serving to said web browser contents of said read-only configuration file member before serving contents of said read-write configuration file.

2. A server system, comprising:  
a first server instance;  
a second server instance;  
said first server instance and said second server instance being resident on a single host system; said first server instance including a first configuration file; and  
said second server instance being an administration server including a second configuration file, said second configuration file being a hardened file not alterable by way of configuration or administration forms by storing administration and configuration directives in a plurality of files including a read-write configuration file member and a read-only configuration file member and serving to said web browser contents of said read-only configuration file member before serving contents of said read-write configuration file.

3. A server system, comprising:  
an administration server instance;  
a global attributes file having one member used by all server instances on a single host server system;  
one or more configuration files, with each server on said single host server system using a particular configuration file member; one or more instance files, with a unique instance file member for each server instance on said single host server system;  
contents of said instance file overriding the contents of said configuration file, and contents of said configuration file overriding contents of said global attributes file; and  
a configuration file for said administration server instance on said single host server system being a hardened file not alterable by way of configuration or administration forms by storing administration and configuration directives in a plurality of files including a read-write configuration file member and a read-only configuration file member and serving contents of said read-only configuration file member before serving contents of said read-write configuration file.

4. The server system of claim 3, further comprising:  
 a browser;  
 said administration server being operable to serve pages to said browser for configuring and managing browser configurable products;  
 said configuration file for said administration server instance restricting usage of said administration server instance to authorized individuals; and  
 said read-only configuration file member being hardened against alteration by any of said pages.
- 10 5. The server system of claim 4, said read-only configuration file member comprising protect, map, pass, and exec directives which cannot be overridden by directives in said read-write configuration file member.
- 15 6. The server system of claim 5, the instance file for said administration file member comprising selected directives, including an autostart directive, which may override directives in said read-only configuration file.
7. The server system of claim 6, said instance file further including number of server jobs, coded character set identifier, server mapping tables, access log file name, error log file name, non-secure port, and secure port directives which may override corresponding directives in said read-only configuration file member.
- 20 8. The server system of claim 7, said global attributes file comprising autostart, minjob, maxjob, ccsid, out ccsid, and in ccsid directives.
- 25 9. A program storage device readable by a machine, tangibly embodying a program of instructions executable by a machine to perform method steps for serving administration and configuration pages to a web browser, said method steps comprising:  
 providing a plurality of instances of servers on a single host system; and  
 one of said servers being an administration server including a configuration file which is not alterable by way of configuration or administration forms by storing administration and configuration directives in a plurality of files including a read-write configuration file member and a read-only configuration file member and serving to said web browser contents of said read-only configuration file member before serving contents of said read-write configuration file.
- 30 10. An article of manufacture comprising:  
 a computer usable medium having computer readable program code means embodied therein for serving administration and configuration pages to a web browser, the computer readable program means in said article of manufacture comprising:  
 computer readable program code means for causing a computer to effect providing a plurality of instances of servers on a single host system, one of said servers being an administration server; and  
 computer readable program code for causing a computer to effect providing for said administration server a configuration file which is not alterable by way of configuration or administration forms by storing administration and configuration directives in a plurality of files including a read-write configuration file member and a read-only
- 35 40 45

- configuration file member and serving to said web browser contents of said read-only configuration file member before serving contents of said read-write configuration file.
11. Method for serving administration and configuration pages to a web browser, comprising the steps of:  
 storing administration and configuration directives in a plurality of files, said files including a global attributes file, a read-write configuration file member, a read-only configuration file member, and an instance file;  
 serving to said web browser the contents of said read-only configuration file member before serving the contents of said read-write configuration file member; and  
 using a first encountered directive for controlling a selected web activity;  
 whereby a directive for a particular web activity stored in said read-write configuration file member cannot override a corresponding directive stored in said read-only configuration file member.
12. A server system, comprising:  
 20 an administration server instance;  
 a global attributes file having one member used by all server instances on said server system;  
 one or more configuration files, with each server on said server system using a particular configuration file member;  
 25 one or more instance files, with unique instance file member for each server instance on said server system;  
 the contents of said instance file overriding the contents of said configuration file, and the contents of said configuration file overriding the contents of said global attributes file;  
 the configuration file for said administration server instance including a read-only member and a read-write member;  
 a browser;
- 30 35 40 45
- said administration server being operable to serve pages to said browser for configuring and managing browser configurable products;  
 said configuration file for said administration server instance restricting usage of said administration server instance to authorized individuals;  
 said read-only configuration file member being hardened against alteration by any of said pages; and  
 said read-only configuration file member comprising protect, map, pass, and exec directives which cannot be overridden by directives in said read-write configuration file member.
13. The server system of claim 12, the instance file for said administration file member comprising selected directives, including an autostart directive, which may override directives in said read-only configuration file.
14. The server system of claim 13, said instance file further including number of server jobs, coded character set identifier, server mapping tables, access log file name, error log file name, non-secure port, and secure port directives which may override corresponding directives in said read-only configuration file member.
15. The server system of claim 14, said global attributes file comprising autostart, minjob, maxjob, ccsid, out ccsid, and in ccsid directives.

\* \* \* \* \*

UNITED STATES PATENT AND TRADEMARK OFFICE  
**CERTIFICATE OF CORRECTION**

PATENT NO. : 6,493,749 B2  
DATED : December 10, 2002  
INVENTOR(S) : Frank V. Paxhia et al.

Page 1 of 1

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Column 42.

Line 28, "file member and serving to said web brower" should be -- file member  
and a read-only configuration file member and serving to said web browser --  
Line 57, "said confifuration file" should be -- said configuration file --

Column 43.

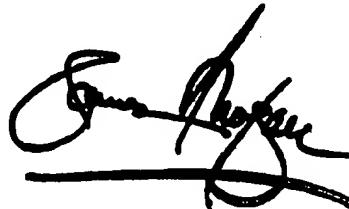
Line 38, after "configuration" delete "file which is not alterable by way of  
configuration"  
Line 43, "condiguration file" should be -- configuration file --  
Line 53, "toeffect" should be -- to effect --

Column 44.

Line 1, "configuratio file" should be -- configuration file --  
Line 2, "member befor" should be -- member before --

Signed and Sealed this

Twenty-seventh Day of May, 2003



JAMES E. ROGAN  
*Director of the United States Patent and Trademark Office*